Universidad
Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería en Informática

PROYECTO FIN DE CARRERA

SISTEMA DE SEGURIDAD Y MONITORIZACIÓN DEL SERVICIO DE TELETRABAJO DE LA UC3M BASADO EN SOFTWARE LIBRE

Autor: Alberto González Piedra

Tutor: Vicente Palacios Madrid

Diciembre 2011

Resumen

Es ingenuo pensar que el mero hecho de poner un equipo a funcionar como servidor es suficiente, siempre y cuando cumpla con la función que se le ha asignado. Lo cierto es que, aunque cumpla con su cometido en el momento de ser instalado, nada asegura que continúe funcionando debidamente a lo largo del tiempo si no cuenta con medidas de seguridad adecuadas.

Este proyecto tiene como objetivo securizar la plataforma de teletrabajo de la Universidad Carlos III. Más concretamente, lograr que los servidores proporcionen un servicio robusto y seguro, utilizando para ello distintos mecanismos, como cortafuegos, herramientas de respaldo o sistemas de monitorización. El funcionamiento de estos mecanismos ha de ser apropiado para el tipo de servicio que ofrecen las máquinas, siendo necesaria la realización de un estudio, un diseño y, finalmente, su implantación.

Para la implantación de servicios de seguridad, no hay herramientas más confiables que aquellas que ofrece el software libre. Gracias a la publicación de su código fuente, podemos asegurarnos que cumplen únicamente con su cometido, sin código que pueda comprometer la privacidad o integridad del sistema. Además, estas herramientas cuentan con un mantenimiento que las hace estar actualizadas con frecuencia, subsanando aquellos errores que puedan presentar. Al ser libres, la utilización de estas herramientas supone un importante ahorro económico, en cuanto al coste de licencias. Todo esto hace que el software libre sea idóneo para el desarrollo de este proyecto.

Palabras clave: Software libre, código abierto, seguridad, cortafuegos, respaldo, monitorización, disponibilidad, virtualización, teletrabajo.

Abstract

It's a wrong idea believing that is enough for a server to accomplish the function it was made for. In truth that a server works at the moment of its setup, doesn't mean it will keep running along the time.

This project aims to offer a safe service for the telework servers of the Universidad Carlos III. To achieve this goal, will be used diferent kind of tools, like firewalls, backup software or monitoring platforms. Those tools have to been configured for the specific features of the server used. That's the reason because it is necessary to make a previous study, design and, at last, its implementation.

If we want to depevelop security services, there aren't more trusted tools than the offered by the free software projects. Thanks to this software's public source code, we can grant that there aren't any malicious codes in these tools. Free software usually counts with a user's and developer's community, who works in its development offering updates periodically to fix most of the software issues detected. Also, free software tools don't need any license to work that could cost us an important invests, making it a cheaper alternative. All that points made free software an ideal alternative for our purposes.

Keywords: free software, open source, security, firewalls, backup, monitoring, availability, virtualization, telework.

Índice

Índice de figuras

Índice de tablas

Capítulo I

Introducción

Casi desde el principio de la existencia de las grandes redes informáticas existe software que vulnera la seguridad de los equipos conectados a ellas. Desde el nacimiento del considerado primer virus informático (Creeper) que ya en 1972 se propagó por la red ARPANET, han surgido numerosas formas de acceder a otros equipos informáticos sin autorización y causar daños o sustraer información, ya sea por motivos económicos o simplemente como reto personal o búsqueda de fama.

Los servidores informáticos son objetivos especialmente interesantes para estos ataques por muchos motivos. Su accesibilidad y disponibilidad los convierten en blancos fáciles y localizables, su hardware hace que cuenten con más recursos que los equipos de sobremesa, ofreciendo un mayor abanico de posibles aplicaciones tras un asalto. Además los servidores suelen ser puntos de convergencia de datos, lugares a donde llega y se almacena gran cantidad de información, como pueden ser datos personales o datos bancarios, de los cuales se puede sacar un gran beneficio de ser obtenidos.

No hay que confiarse y creer que, por el mero hecho de que nuestro servidor este protegido frente a ataques externos (algo que tampoco puede ser absolutamente afirmado, a menos de que se aísle, y por tanto perder su funcionalidad como servidor), es seguro. Como se ha dicho antes, los servidores son puntos de convergencia de la información, una información en casos valiosa, irremplazable o necesaria para el funcionamiento de otros procesos de información. El servidor, a pesar de estar protegido frente a ataques externos, puede ser vulnerable a errores de software, fallos en el hardware, catástrofes físicas como variaciones de tensión, o incendios, o infinidad de causas imposibles de predecir, que pueden provocar la pérdida, temporal o permanente, del servidor, con la consiguiente pérdida de esa información tan importante. A esto hay que añadir la interrumpción del servicio que este ofreciendo ese servidor, que puede ser tan importante o crítico como los datos que almacena.

Salvo catástrofes naturales o físicas, que no son fáciles de prevenir, los problemas originados por el software o hardware sería interesante poder evitarlos, o al menos detectarlos antes de que las consecuencias sean peores, a través de la medición de diversos parámetros de los sistemas informáticos.

En este proyecto se pretende ofrecer una solución práctica a estos problemas planteados, para ser aplicados en los servidores de la Oficina de Software Libre de la Universidad Carlos III de Madrid y, en concreto, los servidores con los que cuenta esta oficina utilizados para proveer máquinas virtuales a trabajadores a distancia: el servicio de Teletrabajo.

1.1. Motivación

Como resultado del proyecto fin de carrera anterior (Gil Bázquez, 2011), se tiene una plataforma de virtualización que sirve máquinas virtuales personales a distintos usuarios predeterminados. Estos usuarios usarán esas máquinas virtuales como si fueran los equipos de su puesto de trabajo, evitando así la necesidad de que el usuario tenga que estar físicamente situado en su puesto de trabajo. Es decir, se ha diseñado y desplegado una plataforma, que denominaremos a partir de ahora servicio de Teletrabajo, que permite a los usuarios trabajar remotamente.

El servicio de Teletrabajo funciona correctamente en un entorno ideal, en el que no hay personas malintencionadas, el software y el hardware no fallan nunca, y tanto los usuarios como administradores no cometen nunca errores. Esta situación no es realista, y con mayor o menor frecuencia surge algún problema que puede poner en peligro la integridad del servicio. En este aspecto el servicio debe tratarse como crítico, dado que, cuando el servicio no está disponible, los teletrabajadores no pueden acceder a él y no pueden desempeñar su trabajo, provocando una pérdida de productividad.

No sólo hay que evitar que los teletrabajadores pierdan el acceso a las máquinas virtuales, sino que hay que proteger los datos que generen como producto de sus sesiones de trabajo. La pérdida de estos datos puede suponer un mal mayor que el simple hecho de no poder trabajar. Si la plataforma no está activa, se pueden llegar a perder unos pocos días de trabajo, que siempre pueden ser recuperados trabajando en un equipo distinto o en el futuro, en cambio, la pérdida de los datos de los teletrabajadores, puede llegar a provocar pérdidas de semanas o meses de trabajo imposibles de recuperar en algunos casos.

También hay que tener en cuenta que el servicio de Teletrabajo está desplegado sobre servidores que, para poder cumplir con su función, han de estar conectados a una red y, a través de ella, a Internet. Esta conexión de red, imprescindible para que los

teletrabajadores puedan acceder a su máquina virtual, es el principal foco de riegos para el sistema y el punto por el que pueden realizar ataques al servidor con diversos objetivos, desde interrumpir el servicio, hasta manipularlo para utilizarlo en otros ataques, pasando por hospedar servicios no autorizados en los servidores.

Por lo tanto, es necesario **proteger** el servicio de Teletrabajo para asegurarse que es capaz de resistir los problemas mencionados, evitando la interrupción del servicio y, en caso de que surja un problema importante, disponer de la capacidad de reacción necesaria para poder restablecer el servicio en el menor tiempo posible.

1.2. Objetivos

El principal objetivo de este proyecto es dotar de fiabilidad y seguridad al servicio de Teletrabajo de la Universidad Carlos III de Madrid, para ofrecer un servicio robusto y de calidad. De tal manera que sea posible asegurar el funcionamiento del servicio ante situaciones adversas, y poder ofrecer a los teletrabajadores una seguridad en sus sesiones de trabajo, de forma transparente para los usuarios.

Puesto que el objetivo principal es abstracto, es necesario cumplir una serie de sub-objetivos que permitan precisar los aspectos de seguridad a cubrir.

En primer lugar, hay que asegurar que únicamente las personas autorizadas tienen acceso al servicio de teletrabajo, de modo que los servidores no queden totalmente accesibles a internet y, por lo tanto, puedan ser el objetivo de ataques maliciosos a manos de terceros. Esto es un aspecto prioritario, ya que **controlar el acceso** a los servidores, no sólo puede evitar un uso ilegítimo y fraudulento de los mismos, o evitar una interrupción del servicio por un ataque destructivo hacia los servidores, sino que podemos evitar el acceso y sustracción de los datos pertenecientes a los teletrabajadores.

Otro objetivo que ofrezca una mayor seguridad al servicio de Teletrabajo es la realización de **copias de seguridad** de las máquinas de los teletrabajadores. Se permite así la recuperación de cualquier máquina virtual ante cualquier imprevisto, como un fallo de funcionamiento debido a la instalación de software inapropiado, errores humanos, software malicioso, etc.

Las copias de seguridad son necesarias pero no suficientes para ofrecer un servicio de calidad, ya que, ¿qué pasa si lo que se estropea no es una máquina virtual, sino el servidor de virtualización entero? En tal caso, ¿cómo se accede a la máquina virtual? ¿cómo se recupera la información de los teletrabajadores si el servidor está tan dañado como para no poder acceder a sus datos? En otras palabras, la **disponibilidad** es otro de

los objetivos a tener en cuenta. A través de la replicación de la configuración y de las copias de seguridad de las máquinas en distintos servidores, se contará con distintas copias de la información necesaria para la continuidad del servicio, accesibles desde las distintas fuentes en caso de que el servidor principal falle.

Para asegurar esa disponibilidad del servicio de Teletrabajo, no basta únicamente con la replicación de servicios o datos, es necesaria una serie de procedimientos, definidos por los administradores, para coordinar los recursos disponibles, subsanar las incidencias y activar el acceso a los servidores de respaldo. Estos **procedimientos** tienen que ser bien definidos, indicando como subsanar cada una de las incidencias que pueden surgir, y cómo minimizar las interrupciones del servicio de Teletrabajo.

Finalizando los objetivos relacionados con la seguridad, y como parte del objetivo de disponibilidad, con objetivo de reducir los tiempos de repuesta en aplicar los procedimientos de restauración, se desea utilizar un sistema que alerte a los administradores cada vez que surja una incidencia. Este sistema deberá estar continuamente realizando mediciones sobre los equipos implicados en el servicio de Teletrabajo, y notificar en el momento de detectar cualquier anomalía. De esta manera se evita que los administradores tengan que estar continuamente supervisando el funcionamiento del servicio, y permite detectar de forma sencilla muchos problemas que no resulten fáciles de verificar por una persona o que resulten muy tediosos.

Junto con los objetivos anteriores, en relación con la fiabilidad y seguridad de la plataforma, es necesario optimizar el servicio de Teletrabajo, para poder así ajustarse a las necesidades de los teletrabajadores y ofrecerles una mejor experiencia de trabajo. Para llevar a cabo este objetivo no basta con parametrizar las máquinas virtuales en cuanto a memoria, disco duro o procesador, y esperar una opinión del usuario. En su lugar, habrá que hacer un estudio de las necesidades de recursos de las máquinas mediante datos reales de su uso, siendo necesaria la implantación de alguna herramienta que permita realizar estas mediciones.

1.3. Estructura del documento

A continuación se exponen los capítulos con los cuales contará el presente documento.

El *Capítulo I Introducción* es el que ocupa estas páginas y pretende poner en situación al lector de las causas que motivaron el desarrollo del proyecto y los objetivos que se desean cumplir a la finalización del proyecto.

Posteriormente se continuará con *Capítulo II Estado de la cuestión* en el que se expondrán los resultados del proceso de documentación previo al desarrollo del proyecto, dando una visión de cada uno de los temas tratados en el proyecto.

Una vez introducidos los temas que se tratarán, y antes de comenzar con el desarrollo en sí del proyecto, se describirán las herramientas software más destacadas que han sido utilizadas en el *Capítulo III Herramientas para la elaboración del proyecto.*

El *Capítulo IV Desarrollo del proyecto* es el centro del documento, al ser donde se detalla cada una de las fases por las que ha pasado el proyecto, desde su análisis y diseño hasta su implantación. Una vez implantado el sistema, también será en este capítulo donde se calculará el tiempo invertido en el desarrollo y se realizará la estimación de costes.

En el *Capítulo V Plan de pruebas* se detallarán una serie de procedimientos para comprobar la funcionalidad del sistema desarrollado y poder comprobar así que cumple la especificación realizada en el capítulo anterior, además de explicar cómo se ha mejorado el rendimiento del sistema una vez éste estaba ya en funcionamiento.

El *Capítulo VI Conclusiones* pretende hacer un resumen de todo lo realizado durante el proyecto para poder comprobar cuántos de los objetivos planteados en el *Capítulo I* se han cumplido, y, de ser así, en qué medida se han cumplido.

A continuación, en el *Capítulo VII Líneas futuras*, se explicarán las ideas que han surgido durante el desarrollo para posibles mejoras y ampliaciones del sistema.

El *Capítulo VIII Glosario* contendrá, ordenados de forma alfabética, la definición de aquellos términos de interés que aparecen en el documento.

Toda la bibliografía consultada y citada durante el desarrollo del documento, así como manuales o sitios web de referencia, están contenidos en el *Capítulo IX Referencias.*

Por último, el documento se finalizará con un anexo en el que se incluyen los scripts complementarios desarrollados.

Capítulo II

Estado de la cuestión

Para establecer las bases sobre las que se apoyará el desarrollo de este proyecto, se intentará dar una visión global de cada uno de los conceptos que intervienen en el proyecto.

Aunque el objetivo del proyecto es dotar de seguridad a un servicio ya existente, los temas a tratar no se limitarán exclusivamente a la seguridad, haciéndose mención a los conceptos relacionados con el servidor original: **Teletrabajo** y **Virtualización** que se corresponden con los dos primeros puntos. Estos puntos pretenden ofrecer un breve resumen, al no ser los temas centrales de este proyecto, pudiendo obtenerse una visión más amplia en el proyecto previo (Gil Bázquez, 2011).

El siguiente punto tratará sobre el **Software libre**, que es de vital importancia para el proyecto al ser un requisito que condicionará desde la elección de herramientas hasta la solución implementada.

Por último, se pasará a hablar sobre el tema central: la **Seguridad**. Aquí se tratará sobre el control de acceso y la herramienta utilizada para conseguirlo, los cortafuegos, seguido de una introducción sobre las copias de seguridad y el desarrollo de distintas técnicas para llevarla a cabo. El último punto, **Monitorización**, se ha separado del de seguridad por el enfoque distinto que se le ha dado, ya que no sólo se definirá que es la monitorización, sino que además incluirá las características de algunos productos libres existentes para llevarla a cabo.

2.1. Teletrabajo

El origen del teletrabajo se remonta al año 1970, cuando la empresa Hewlett-Packard se dio cuenta de las horas que invertían sus empleados en desplazarse hasta su puesto de

trabajo. El desplazamiento provocaba un cansancio en los empleados que provocaba un menos rendimiento en su trabajo. La solución fue establecer un sistema con el que los empleados pudieran trabajar directamente con ficheros en el propio servidor (Cimarra Cardenal, 2005).

El teletrabajo surge como una alternativa con diferentes ventajas:

- Conciliación de la vida familiar con la laboral.
- Ahorro de tiempo para los empleados.
- Oportunidades de trabajo para discapacitados.
- Reducción de costes en inmuebles para las empresas.
- Salud laboral.
- Reducción del tráfico y de la emisión de agentes contaminantes (aunque aumenta el consumo energético en los hogares de los trabajadores).

El teletrabajador obtiene un beneficio en su salud al no necesitar desplazarse hasta su puesto de trabajo, empezando por las dietas. Al comer en casa, puede seguir una alimentación más saludable, además comer en casa conlleva un ahorro económico respecto a tener que comer fuera de casa. Al no estar expuestos a la polución o las inclemencias del tiempo reduce las probabilidades que una persona padezca una enfermedad. Otra ventaja para la salud es disfrutar de un ritmo de vida más pausado, al no estar sometido a las prisas y estrés que suponen el desplazamiento, trabajar en el hogar también reduce el estrés de los empleados al estar en un entorno más agradable, sin el ajetreo de la oficina, o sin las presiones de los superiores. Por último, el ahorro de tiempo, hace que se invierta el tiempo ganado en hacer ejercicio, que repercute positivamente en la salud del teletrabajador (Cimarra Cardenal, 2005).

Las empresas por su parte obtienen beneficios no solo en cuanto a la productividad de los empleados, que se estima que se incrementa entre un 10% y un 40%(Kurland y Bailey, 1999; Bailey y Kurland, 2002), sino que se ahorra dinero en inmuebles al no necesitar oficinas o plantas tan grandes para albergar tantos empleados, y en consecuencia el consumo energético. Las empresas, mediante el teletrabajo, también consiguen que retener a los empleados no solo por las condiciones ofrecidas, sino que el trabajador no se ve obligado a abandonar su puesto de trabajo por motivos como puede ser la distancia física, o tener a otra persona a su cargo. De esta manera se evita la pérdida de empleados con una importante especialización, reduciendo los costes de búsqueda y contratación de un nuevo empleado(Blanco Romero, 2005).

2.2. Virtualización

El concepto virtualización lleva de la mano el de máquina virtual, y puede definirse como el proceso de encapsular una unidad de computación (equipo completo, sistema operativo y programa) dentro de un equipo físico real de forma transparente (Galán Márquez y Fernández Cambronero, 2004).

La máquina virtual se refiere a los componentes emulados dentro de otra máquina (que denominaremos anfitriona). La máquina virtual a efectos prácticos, debería de ser equivalente, en cuanto a comportamiento, a un equipo real, exceptuando las limitaciones propias como son una menor disponibilidad de recursos y un aumento en los tiempos de respuesta con los dispositivos. Una máquina virtual es una unidad de computación independiente, no afectando su funcionamiento al de otras posibles máquinas virtuales alojadas en la misma máquina física. Además el rendimiento de la máquina virtual debería ser similar al del hardware de la máquina anfitriona (Fuertes Díaz y López de Vergara Méndez, [s.f.]).

En la informática, virtualización se refiere a la abstracción de recursos de un ordenador. En general, la virtualización aporta grandes ahorros a determinados sectores de la informática, ya que es resulta más barato mantener un único servidor físico que soporta varias máquinas virtuales que ofrecen diversos servicios, a tener que mantener un gran número de máquinas independientes dedicadas a un servicio cada una (Jones y González, 2008).

En esencia, la virtualización trata de reemplazar un hardware, y sustituirlo por otro más potente, que por mediación de software cubra los mismos servicios que el hardware original. Esto permite un mayor aprovechamiento del hardware, y una mayor flexibilidad, permitiendo reasignar recursos en cualquier momento de una máquina a otra en momentos de necesidad (Arias Chaves, 2008).

La virtualización es posible hoy en día gracias a la gran capacidad que poseen los ordenadores. Los procesadores están prácticamente la mayor parte del tiempo ociosos, ya que solo están activos cortos periodos de tiempo en los que atienden las peticiones que reciben, además, casi todos los procesadores modernos instalados en los servidores están preparados para ejecutar varios sistemas operativos simultáneamente. Respecto a la capacidad de almacenamiento y memoria RAM, los servidores suelen ofrecer una capacidad muy superior a la de los ordenadores de hace pocos años y resulta barato realizar ampliaciones para albergar un mayor número de máquinas virtuales.

En un sistema virtualizado, existen unos componentes software denominados hipervisores, que se sitúan entre el hardware físico y las aplicaciones de negocio, y ofrecen

réplicas virtuales de los dispositivos físicos. Las aplicaciones de la máquina virtual, dejan de comunicarse directamente con el hardware físico de una máquina, para pasar a comunicarse con estos hypervisores.

El hypervisor o VMM(Virtual Machine Monitor), se encarga de la gestión de recursos (CPU, RAM, almacenamiento, etc.), permitiendo al usuario la definición de un hardware virtual concreto que sea compatible con el sistema operativo instalado en la máquina virtual. Además, es esta capa software la que permite la asignación dinámica de recursos entre las máquinas virtualizadas (Arias Chaves, 2008; García Calahorro, 2009).

2.3. Software Libre

Para dar una visión que permita comprender la filosofía del software libre, se va a comenzar relatando una breve historia sobre los orígenes del software, software propietario, y lo que motivó a la aparición del software libre.

Una vez explicado los orígenes del software libre y dar una idea de en qué consiste, se pasará a definirlo desde el punto de vista de los dos movimientos principales que lo promueven y las herramientas que utilizan para asegurar la libertad del software.

En el último punto se listarán una serie de característica en forma de ventajas y desventajas que sirvan para comparar los dos puntos de vista del software, propietario y libre.

2.3.1. Historia del software libre

Durante los años 1960 el campo de la informática estaba dominado por los grandes ordenadores instalados en empresas y centros gubernamentales. Durante esta época, el mayor fabricante de ordenadores era IBM, y cuando comprabas un ordenador, lo que realmente comprabas era el hardware, estando el software incluido como un acompañante. Mientras se pagase el contrato de mantenimiento, se podía acceder al catálogo de software del fabricante. El software obtenido del fabricante venia siempre acompañado de su código fuente y generalmente sin ningún tipo de restricción práctica, incluso en muchos casos, únicamente se distribuía el código fuente. Además no era común pensar que el software pudiese ser comercializado de forma separada al hardware.

Se puede empezar a hablar de software libre en estos años por el conocimiento del código fuente del software, su ausencia de restricciones y la aparición de grupos de

usuarios que contribuían a la compartición del software (como el grupo SHARE de usuarios de IBM y DECUS de usuarios de DEC).

En 1970 IBM empezó a comercializar por separado el software. Por tanto los usuarios dejaron de recibir el software al adquirir el hardware. El software se empezó a ver como algo con valor intrínseco y, como consecuencia, se comenzó a restringir el acceso y la compartición del software con medidas tanto técnicas como legales.

En los años posteriores, pese a tendencia de crear software privativo, se originaron iniciativas que mostraron características de lo que más adelante se conocería como software libre. Entre estas iniciativas se encuentran TeX, SPICE o Unix. En concreto SPICE fue desarrollado en la Universidad de Berkeley, por Donald O. Pederson, en 1973. SPICE fue puesto en dominio público, permitiendo de esta manera ser modificado o redistribuido sin restricciones, popularizándose por ello en diversas universidades, y, llegando incluso a permitir su modificación y comercialización del software como un producto privativo.

Por su parte Unix (uno de los primeros sistemas operativos portables), fue desarrollado hacia 1972 en los laboratorios Bell de AT&T. En los años 1973 y 1974, Unix se distribuyó por varias Universidades y centros de investigación con una licencia que permitía su uso para fines académicos. Aunque había restricciones en su licencia sobre la redistribución del software, entre las organizaciones que disponían de una licencia de Unix, se comenzó a utilizar un funcionamiento similar al que se tiene hoy en día en las comunidades de software libre. Los que tuvieron acceso a Unix, en seguida comenzaron a estudiarlo y a desarrollarlo implantando mejoras y expandiéndolo, así surgió una comunidad de desarrolladores en torno al CSRG (Computer Science Research Group) de la Universidad de California en Berkeley.

A principios de 1984, Richard Stallman, abandonó su trabajo en AI Lab del MIT. para comenzar el proyecto GNU. Su idea era construir un sistema software completo y de propósito general completamente libre. El sistema, y proyecto que lo desarrollaría, fue denominado GNU (GNU's Not Unix). A pesar de que se incluyó en el proyecto mucho software ya existente, como TeX o X Window, había mucho software que construir. Así Richard Stallman comenzó a escribir un compilador de C (GCC) y un editor de texto (Emacs), ambos en uso hasta estos días.

Ilustración 1 El ñu mascota del proyecto GNU

Richard Stallman estaba preocupado por las libertades de los usuarios de su software, interesándose no solo en que los usuarios recibieran los programas directamente desde el proyecto GNU, sino que también pudiesen recibirlo a través de redistribuciones y posibles modificaciones, teniendo las mismas libertades que tenía el software en su origen. Para asegurar esta libertad escribió la licencia GPL, que probablemente fue la primera licencia que aseguraba la libertad del software. Al mecanismo genérico utilizado en las licencias GPL para conseguir estas garantías, fue bautizado por Richard Stallman como copyleft.

Por aquel entonces Richard Stallman fundó la Free Software Foundation (FSF) para conseguir fondos que dedicar al desarrollo del software y proteger la libertad del mismo. También plasmó sus fundamentos éticos en el manifiesto de GNU y Why Software Should Not Have Owners.

Ilustración 2 Logotipo de la Free Software Foundation

El proyecto GNU lo formaban pequeños grupos de personas (generalmente voluntarios), encargados del desarrollo de alguna de las herramientas necesarias para el sistema, que posteriormente serían integradas en el sistema GNU. Puesto que no había aún una gran expansión de Internet para la distribución del software, la Free Software Foundation grabó unas cintas con el software para su venta a los usuarios, siendo también la primera organización en obtener beneficios (aunque limitados) con la creación de software libre.

Seis años después (a principios de la década de 1990), el proyecto GNU ya estaba a punto de terminar un sistema completo similar a Unix, salvo por la falta de un componente: el núcleo del sistema. A pesar de la falta de un núcleo, las aplicaciones del proyecto GNU habían contado con una gran expansión y popularidad (sobre todo entre las universidades y profesionales) gracias a su fama de estabilidad y calidad.

Desde 1973 hasta 1980 el CSRG de la Universidad de California fue uno de los centros donde más se desarrolló el sistema Unix. Se portaron aplicaciones, se mejoró el núcleo del sistema y se le añadió nueva funcionalidad. En la década de 1980, DARPA financió el desarrollo y la implementación de TCP/IP que se desarrolló en esta misma Universidad. Fue tal la expansión de Unix por el CSRG que las empresas comenzaron a basar sus sistemas operativos en esta versión del sistema, en lugar de la original de AT&T.

Sin embargo utilizar el código del CSRG requería la adquisición de una licencia de Unix de AT&T, que con el tiempo se volvió más difícil y cara de conseguir. Por este motivo, en 1989, el CSRG liberó la parte de código de Unix relacionada con TCP/IP denominándola Network Release 1 (Net-1). Net-1 se liberó con la licencia BSD, que es la otra gran licencia de software libre. Esta licencia permitía la libertad de distribución del código y su incorporación en productos que fueran privativos.

El éxito de Net-1, hizo que Keith Bostic propusiera reescribir todo el código que era propiedad de AT&T, para lo cual solicitó la colaboración del público para reescribir sus aplicaciones, mientras se reescribía el código del núcleo. De esta manera, en junio de 1991 se distribuyó la Networking Release 2 (Net-2) que era prácticamente un sistema Unix completo distribuido bajo la licencia BSD.

Seis meses más tarde, Bill Jolintz escribió el código restante para portar Net-2 a la arquitectura i386, distribuyéndolo a través de Internet bajo el nombre 386BSD. 386BSD fue la base para la familia de sistemas BSD que surgieron y se siguen utilizando actualmente, como son NetBSD, FreeBSD y OpenBSD.

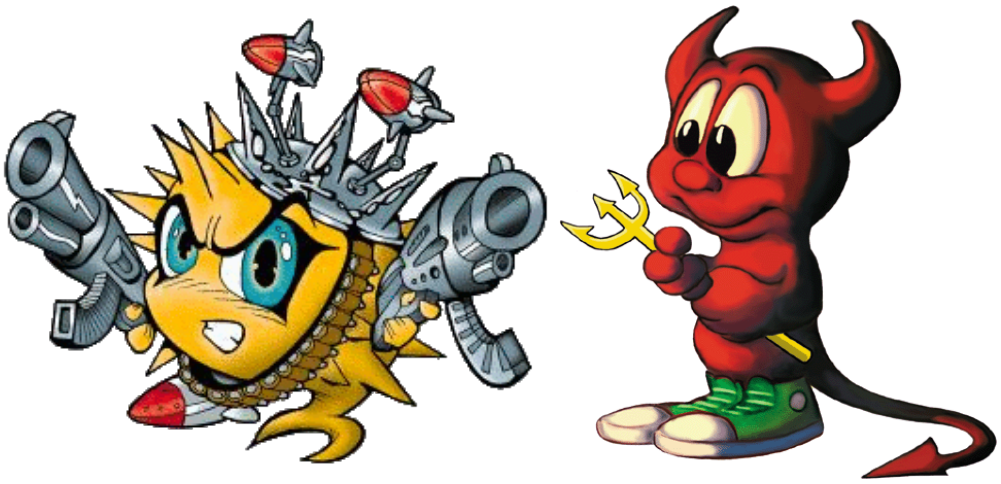

Ilustración 3 Puffu y Beastie las mascotas de OpenBSD y FreeBSD

En septiembre de 1991 un estudiante finés de 21 años llamado Linus Torvalds libera la primera versión (0.01) del núcleo de Linux con la intención de crear un sistema libre similar a Minix. En marzo de 1994 se completó la primera versión estable (1.0) del núcleo. Durante este periodo de tiempo, cientos de desarrolladores integran alrededor de este núcleo todo el software de GNU, XFree y otros programas libres. La gran diferencia que presentaba frente a BSD es que todo el software integrado alrededor de Linux utilizaba la licencia GPL.

Ilustración 4 Tux, la mascota de Linux

Entre los desarrollos en torno al núcleo de Linux, aparece el concepto de distribución en 1992. Las distribuciones pretenden ofrecer un sistema GNU/Linux listo para usarse y basándose todas en el mismo software. Además de paquetes precompilados disponibles para su uso, las distintas distribuciones ofrecen herramientas para la gestión de estos paquetes: instalación, desinstalación o actualización. Las distribuciones también suelen ofrecer medios para la instalación de

las mismas en un equipo y ofrecen herramientas para la gestión del sistema operativo (Seoane Pascual; González Barahona y Robles, 2007).

Entre las distribuciones de GNU/Linux actualmente disponibles se encuentran algunas de las más populares históricamente:

- Red Hat Linux, mantenida por la empresa del mismo nombre, fue una de las primeras distribuciones en comercializar productos para empresas basados en software libre. Con el tiempo se ha dividido en dos proyectos: Red Hat, que es una versión de pagos destinada a empresas; y Fedora, la versión totalmente libre mantenida por la comunidad. Algunas distribuciones que surgieron de Red Hat son Mandrake (actualmente ha evolucionado a Mandriva) o CentOS.
- Debian, una distribución totalmente de código abierto e independiente de cualquier empresa, encargándose del mantenimiento su propia comunidad de desarrolladores voluntarios. A partir de Debian han surgido otras distribuciones como Knoppix que fue la primera distribución en poder arrancarse desde un disco óptico o el popular Ubuntu que añade software privativo al suministrado por Debian.
- openSUSE, nacida tras la compra por parte de Novell (la actual propietaria de Unix) de the SUSE Linux company encargada del desarrollo de las distribuciones SuSE Linux (S.u.S.E. Linux en sus orígenes) que se vendían en tiendas con una gran documentación. La principal diferencia de openSUSE es que utiliza únicamente software 100% open source.

2.3.2. ¿Qué es el software libre?

El término "libre" en el software libre no tiene connotaciones económicas como mucha gente cree (free en inglés), sino que hace referencia a la libertad del usuario.

El software libre pretende asegurar las libertades del usuario de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. En concreto los usuarios de programas tienen **las cuatro libertades esenciales**:

Libertad 0. La libertad de ejecutar el programa, para cualquier propósito.

Libertad 1. La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.

Libertad 2. La libertad de redistribuir copias para que pueda ayudar al prójimo.

Libertad 3. La libertad de distribuir copias de sus versiones modificadas a terceros. Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa puede ser considerado software libre si ofrece todas estas libertades. En este caso cualquiera tiene derecho a distribuir copias del software ya sea en su estado original o modificadas, y gratuitamente o cobrando un coste por la distribución, sin necesidad de obtener una autorización o pagar tarifas para hacerlo.

La distribución del software debe hacerse tanto en forma de código fuente y en forma binaria para poder ser ejecutado de una manera sencilla, no obstante se permite como excepción no distribuir los ficheros binarios en caso de que no sea posible generarlos (por ejemplo lenguajes interpretados).

Además la distribución del código fuente, para poder cumplir las libertades 1 y 3, debe ser claro. Aquel software con código fuente ofuscado, no puede ser considerado software libre (Free Software Foundation, 2011a).

2.3.2.1. Copyleft

Copyleft es la herramienta con la que se asegura (mediante su implementación en la licencia GPL) que el software libre sigue siendo libre tras sucesivas modificaciones. Incluso asegurando que cualquier software que utilice algún componente libre siga siendo en su conjunto software libre (Stallman, 2004)

La forma más sencilla de liberar software sería ponerlo bajo dominio público (sin derechos de autor). Permitiendo así que cualquiera tengo acceso al software para modificarlo o compartirlo. Sin embargo no se puede asegurar que se mantenga una actitud cooperativa y, que alguien realice modificaciones, convirtiéndolo en software privativo. Cuando el software modificado y privatizado es distribuido, las personas que reciben ese software ya no poseen las libertades que ofrecía el software en su origen al haber sido eliminadas por el intermediario.

Utilizando una licencia *copyleft*, se protege el software (frente a publicarlo bajo dominio público) asegurando que siempre mantenga las libertades, ya que cualquier modificación de software libre, o uso del mismo como componente de un software mayor, hace automáticamente que el producto final sea libre.

Intentar privatizar el software modificado para obtener beneficios económicos sería ilegal, si este está protegido por el *copyleft*, teniendo como alternativas o bien desechar el código o liberarlo para contribuir con el software libre.

Puesto que el *copyleft* es un concepto, hay que recurrir a una implementación del concepto que cumpla con su función. Esto es la *Licencia Pública GNU* (GPL), con la cual se deben registrar los derechos del autor del software (copyright). De esta manera se utiliza el copyright, que usualmente se utiliza para declarar derechos reservados de un producto, para asegurar su libertad y protegerla legalmente (Free Software Foundation, 2011b).

2.3.2.2. Open source

La diferencia entre el software libre y el código abierto es principalmente el punto de vista ideológico. El movimiento de software libre y la Free Software Foundation defienden la libertad del software por motivos éticos, mientras que el movimiento de código abierto promovido por la Open Source Initiative tiene motivaciones prácticas.

Ilustración 5 Logotipo de la Open Source Initiative

Para que el software puede ser considerado de código abierto (open source) no basta con que sencillamente el código fuente esté disponible, sino que debe de cumplir las siguientes características (Open Source Initiative, 2003):

1. **Redistribución libre**: la licencia no impedirá la venta o el ofrecimiento como componente de una distribución de software. En ningún caso la licencia requerirá el pago de derechos de autor.

2. **Código fuente**: el código fuente ha de distribuirse junto al código compilado. Si el producto no se distribuye con el código fuente debe de proporcionarse un medio para conseguirlo, preferiblemente por internet. De aplicarse tasas la distribución del código fuente estas deben cubrir únicamente los costes de distribución.
3. **Trabajos derivados**: la licencia ha de permitir realizar modificaciones del software, y permitir que ese software modificado sea distribuido con los mismos términos de licencia que el código original.
4. **Integridad del código fuente del autor**: la licencia puede impedir que el código fuente sea distribuido de una forma modificada, si se pueden distribuir parches para ese código, de manera que sea posible reconstruir el software modificado en tiempo de construcción.
5. **No a la discriminación de personas o grupos**: la licencia no debe impedir el uso de los programas sea cual sea el uso que se le vaya a dar.
6. **Distribución de la licencia**: los derechos de un programa deben poder aplicarse a la totalidad de este sin restricciones de una segunda licencia de alguna de las partes del código.
7. **La licencia no tiene que ser específica de un producto**: los derechos adjuntos a un programa no son dependientes a que ese programa forme parte de una distribución de software mayor. Si ese programa es extraído de la distribución de software y distribuido, debe proporcionar los mismos derechos que ofrecía la distribución entera.
8. **La licencia no tiene que restringir a otro software**: si el software licenciado se distribuye junto con otro software, no tiene por qué forzar a que ese otro software sea de código abierto.
9. **La licencia debe ser independiente de la tecnología**: la licencia no debe preferir una tecnología, estilo o interfaz.

2.3.3. Ventajas y desventajas del software libre

A continuación se realizará una comparativa sobre las características que diferencian al software libre del software propietario. La comparativa se realizará

basándose en las ventajas y desventajas que aportan utilizar un tipo u otro de software según (Culebro Juárez; Gómez Herrera y Torres Sánchez, 2006).

2.3.3.1. Ventajas del software libre

1. **Bajo coste de adquisición y libre uso.** El software no se vende de la misma manera que pueda ser vendido cualquier otro producto, sino que se adquieren licencias que definen los usos que se le puede dar al programa en cuestión. Pero el coste del software no se limita a la adquisición de la licencia, sino que ajustarlo, mantenerlo y operarlo conlleva unos costes añadidos. El software libre no requiere de una inversión para la adquisición de licencia, reduciendo los costes, algo que puede atraer al usuario al no verse limitado económicamente para utilizar el software. Las libertades del software libre es una de las características que más puede atraer a los nuevos usuarios. Estas libertades, que permiten la libre modificación del software, promueven que una gran cantidad de usuarios y desarrolladores detecten y corrijan problemas, mejoren el funcionamiento, añadan funcionalidades o adapten el software a situaciones específicas. El resultado obtenido es un software que tiende a ser muy eficiente, robusto y diverso.
2. **Innovación tecnológica.** El software libre, tiene como objetivo compartir la información y trabajar de forma cooperativa: el conocimiento pertenece a la humanidad. Por lo tanto son los usuarios los que deciden hacia qué dirección debe encaminarse el software, votar que problemas subsanarse y proponer la nueva funcionalidad de los programas.
3. **Requisitos de hardware menores y durabilidad de las soluciones.** No se puede generalizar que el software libre tenga menores necesidades de hardware que el software propietario (por tanto más barato de implantar). Pero sin embargo se puede adaptar el software a las necesidades de hardware que se presenten, como por ejemplo eliminar la interfaz gráfica en servidores. Por otra parte en el software propietario, la empresa que lo desarrolla puede en cualquier momento dejar de dar soporte a un determinado hardware que considere antiguo. El software libre por el contrario no depende de una empresa que decida a que hardware se dará soporte, sino que es la

comunidad quien decide cuando dejar de utilizar y dar soporte a un determinado hardware.

4. **Escrutinio público.** El modelo de desarrollo del software libre sigue un método en el que trabajan voluntarios de forma cooperativa y coordinada a través de Internet. Puesto que el código fuente esta visible por todos, es frecuente la detección y notificación de fallos detectados en el código. La notificación es igualmente pública y cualquiera puede aportar una solución, siendo frecuente que el tiempo de respuesta sea muy rápido. Si se compara con el software propietario, debido al secreto del código, únicamente la compañía propietaria del software es quien puede corregir los problemas, haciendo que la solución no se aporte tan rápidamente (si llega a solucionarse).

5. **Independencia del proveedor.** La libertad y conocimiento del código, hace que cualquier empresa o profesional, con conocimientos suficientes, pueda seguir ofreciendo desarrollo o servicios para la aplicación. En el mundo del software propietario, solo el propietario de la aplicación puede ofrecer todos los servicios, creándose una dependencia entre el cliente y el fabricante, ya que el cliente queda atado al software del fabricante. Esto es especialmente cierto cuando el software privativo almacena los datos del cliente de forma desconocida, y entonces el cliente para poder seguir accediendo a los datos almacenados tiene que seguir utilizando el software de ese fabricante.

6. **Datos personales, privacidad y seguridad.** Puesto que el código fuente del software libre es público, se puede conocer exactamente qué es lo que hace el software, y en caso de tratar con datos, como realiza el procesado de los mismos y si llega el caso como los almacena, de manera que sería posible recuperar esos datos almacenados. El software propietario por su parte, se comporta como una caja negra, si se han de introducir datos, no se sabe cómo son procesados, ni como son almacenados. Suponiendo el caso de que se trabaje con datos personales o bases de datos con información de otras personas, no es posible asegurar la absoluta confidencialidad de esos datos al no conocer el proceso con el que se tratan los datos.

7. **Adaptación del software.** El software propietario se vente normalmente en forma de paquete estándar, que muchas veces no se

adapta totalmente a las necesidades específicas de empresas o de administración. Al disponer de su código fuente, el software libre permite la total personalización del programa para adaptarse a las necesidades específicas que se tengan. La personalización es un área importante en el que el software libre puede responder mucho mejor que el software propietario.

2.3.3.2. Desventajas del software libre

1. **La curva de aprendizaje es mayor.** Para alguien totalmente ajeno a la informática a la que se le ponga ante un sistema con una interfaz de usuario de Windows o una interfaz Gnome o KDE, probablemente tarde lo mismo en aprender a usar una que otra. En cambio en la sociedad actual en la que el software propietario es una imposición en muchos casos, es ese el software que sabe utilizar la población, resultándole más difícil aprender a utilizar el software libre.
2. **El software libre no tiene garantía proveniente del autor.** No solo el software libre no se responsabiliza de posibles daños ocasionados por el uso del software (pérdida de datos, mal uso del hardware), si no que no ofrece garantías sobre su funcionamiento.
3. **Se necesita dedicar recursos a la reparación de errores.** Respecto al software propietario en el que no se pueden reparar los errores y hay que esperar a que los solucione el fabricante.
4. **La mayoría de la configuración del hardware no es intuitiva.** Se requieren conocimientos sobre el sistema operativo y el hardware para poder configurarlo correctamente, aunque suele haber una extensa documentación que explique cómo hacerlo.
5. **Únicamente los proyectos importantes y de trayectoria tienen buen soporte, tanto de los desarrolladores como de los usuarios.** Existen muchos proyectos pequeños que carecen del compromiso suficiente de los desarrolladores para ser implementados de manera eficiente. El software con mejor soporte no obstante cubre el 90% de las necesidades informáticas del usuario promedio.
6. **El usuario debe tener nociones de programación.** A pesar de que hoy en día existen herramientas gráficas que ayudan a la administración del sistema, gran parte recae en la automatización de

tareas mediante algún lenguaje interpretado como *Bash*, *Perl* o *Python*.

7. **La diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas con un mismo fin, etc., pueden crear confusión en cierto número de personas.** Aunque también hay quien opina que esto es una fortaleza ya que de esta manera hay una distribución optimizada para cada uso diferente.

2.3.3.3. Ventajas del software propietario

1. **Control de calidad.** Por lo general los productores de software tienen un departamento de control de caridad que realiza exhaustivas pruebas al software.
2. **Recursos a la investigación.** Se destina una parte de los recursos a la investigación sobre los usos del software.
3. **Personal altamente capacitado.** A diferencia del software libre en que los desarrolladores son voluntarios, en el software propietario suelen ser programadores profesionales con experiencia.
4. **Uso común por los usuarios.** Los programas propietarios están tan extendidos que es sencillo encontrar a alguien que los sepa utilizar.
5. **Uso para aplicaciones muy específicas.** Mucho software (sobretodo profesional) esta tan especializado en un determinado sector, que puede que únicamente lo produzca una compañía.
6. **Difusión de publicaciones acerca del uso y aplicaciones del software.** Existe una gran cantidad de publicaciones, ampliamente difundidas, que documentan, facilitan y promocionan el uso de software propietario.

2.3.3.4. Desventajas del software propietario

1. **Cursos de aprendizaje costosos**. Para usar eficientemente el software propietario es necesario asistir a cursos.
2. **Secreto del código fuente**. El código fuente del software solo lo conoce la empresa que lo produce, por lo que un usuario no sabe

exactamente qué es lo que hace determinado componente software siendo arriesgado su uso al poder obtener resultados impredecibles. El desconocimiento del código además evita la corrección de errores.

3. **Soporte técnico ineficiente**. En la mayoría de los casos el soporte técnico es insuficiente o tarda demasiado en ofrecer una respuesta satisfactoria.
4. **Ilegal o costosa la adaptación de un módulo del software a necesidades particulares**. La modificación personal del software para que se adapte a una necesidad suele ser ilegal ya que la licencia de este tipo de software no lo permite. De hecho las modificaciones tienen un coste muy elevado pues hay que delegarlas en la compañía que provee el software.
5. **Derecho exclusivo de innovación**. La innovación es derecho de la compañía fabricante. Si alguien tiene una idea sobre una mejora del software, o bien vende la idea a la compañía o escribe un programa nuevo que la incorpore.
6. **Ilegalidad de copias sin licencia para el efecto**. Es ilegal la copia y uso de software propietario si no se ha comprado una licencia que te permita hacerlo.
7. **Quedar sin soporte técnico**. Si desaparece la compañía proveedora de determinado software, el soporte de ese software desaparece y dejarán de aparecer nuevas versiones (a pesar de que el usuario haya adquirido una licencia de ese software).
8. **Cese de una línea de software.** Las compañías pueden dejar de producir determinado software por motivos como que deje de ser rentable, o modifiquen su modelo de negocio. Provocando que ese software no vuelva a ser modificado nunca más.

2.4. Seguridad

Cuando un equipo se conecta a una red informática surgen tres áreas de riesgo. En primer lugar, se incrementa el número de puntos desde el que se puede iniciar un ataque contra cualquier componente de la red. Cuando un equipo no está conectado a ninguna red, es imprescindible que el atacante tenga acceso físico al equipo, en cambio, en un sistema en red, cada equipo que pueda enviar información a la víctima puede ser utilizado

para atacar. Algunos servicios de red, necesitan estar públicamente y continuamente accesibles desde la red para que puedan desempeñar su función. Haciéndolos susceptibles a ser atacados por cualquier equipo conectado a internet y ser el objetivo de ataques regulares.

En las máquinas aisladas, se puede considerar que cualquier actividad que desempeñan es segura, ya que ejecutan código de la memoria, que han cargado previamente de su almacenamiento secundario. Estos datos están bien protegidos frente a modificaciones u observaciones al ser tratados y transferidos por componentes de confianza. Sin embargo, cuando los datos se transfieren a través de la red. La información es retransmitida por equipos que están totalmente fuera del control del receptor. La información podría ser leída, almacenada o modificada para luego ser retransmitida al receptor legítimo.

Por último, la tercera área de riesgo se refiere al incremento de servicios de autenticación que surgen en un equipo conectado a la red, cada uno de estos servicios es ofrecido por un programa distinto que puede presentar errores en su programación que de ser aprovechados, podrían comprometer la totalidad del sistema (García Alfaro, 2004).

Podemos considerar que la seguridad comprende los mecanismos mediante los cuales evitar estas situaciones. Para ello, en un primer punto se explicarán las herramientas encargadas del control de acceso de un sistema, que prevendrán los accesos maliciosos al sistema. El otro concepto sobre el que se hablará referente a la seguridad son las copias de seguridad, que permitan recuperar el sistema una vez haya sido comprometido, o haya dejado de funcionar debido a algún fallo.

2.4.1. Control de acceso con cortafuegos (firewall)

Un firewall es un dispositivo que filtra tráfico entre al menos dos redes. Puede tratarse tanto de un dispositivo físico o software que funcione sobre un sistema operativo. Un firewall se puede ver como una caja con dos o más interfaces de red, en el que se establecen unas reglas de filtrado. Con estas reglas decide qué conexiones dejará pasar entre una interfaz y otra, o incluso puede realizar modificaciones sobre las conexiones (NAT).

Hoy en día los firewall son un hardware específico con un sistema operativo y dos tarjetas de red, que filtra el tráfico TCP/UDP/ICMP/IP y decide si un paquete pasa, se modifica o se descarta.

En sistemas Linux, el filtrado de paquetes se lleva a cabo por el módulo del núcleo Netfilters, e iptables que es la parte que utiliza el usuario.

El filtrado de paquetes permite aplicar varios criterios a los diferentes paquetes. Se pueden aplicar a paquetes de entrada, de salida o paquetes que atraviesen el firewall. Las decisiones tomadas se pueden basar en la dirección de la que vienen los paquetes, la dirección a la que van, el puerto al que están dirigidos. Se pueden aplicar reglas distintas en función del protocolo al que pertenezca el paquete (TCP, UDP o ICMP).

Un firewall no es un servidor como cualquier otro, ya que debe encontrarse bajo unas condiciones de seguridad física adecuadas, tales como su acceso restringido. Además, es recomendable que no haya otros servicios, como ftp, telnet, etc., ejecutando en el mismo equipo (Ibarra Lemas, 2006).

Se pueden diferenciar tres grandes categorías en las que clasificar los firewalls: Encaminadores con filtrado de paquetes, pasarelas a nivel de aplicación y pasarelas a nivel de circuito.

2.4.1.1. Encaminadores con filtrado de paquetes

Este tipo de dispositivos, encaminan el tráfico TCP/IP basándose en una serie de reglas de filtrado que determina que paquetes se encaminarán a través del dispositivos y cuales se descartarán.

Ilustración 6 Esquema de encaminador con filtrado de paquetes

En la *Ilustración 6 Esquema de encaminador con filtrado de paquetes* se ve un esquema del dispositivo conectado a dos redes: morada y roja. Las reglas de

filtrado son las encargadas de determinar que paquetes pueden pasar de la red externa a la red interna y viceversa.

Los encaminadores con filtrado de paquetes, trabajan a nivel de red, consultando las cabeceras del protocolo (TCP, UDP, etc.) para determinar cómo filtrar el paquete. Los parámetros que puede tener en cuenta son: las direcciones, tipo de protocolo y sus indicadores, puertos de origen y destino, contenido del paquete o tamaño del paquete.

Las reglas, están organizadas en conjuntos de listas que siguen una determinada política por defecto (aceptar todos los paquetes, rechazar todos los paquetes). De tal manera que si llega un paquete que no pueda ser aplicado a ninguna de las reglas, se realizará con ese paquete la acción que indique la política por defecto.

Las políticas permiten gestionar el firewall desde dos puntos de vista alternativos. Una política de denegación por defecto es más costosa de mantener, puesto que es el administrador quien debe indicar explícitamente todos los servicios que deben permanecer abiertos. En cambio, una política de aceptación por defecto simplifica la administración, ya que únicamente hay que indicar que conexiones no se permitirán, pero a cambio se incrementa el riesgo de ataque contra la red al ser difícil asegurar que se controlan todas las conexiones peligrosas.

Utilizar un firewall mediante encaminador con filtrado de paquetes es una solución económica, ya que suelen ser construidos sobre un hardware ya disponible. Además ofrece un alto rendimiento ante altas cargas de tráfico de red, y permite la implantación de la mayoría de las políticas de seguridad deseadas.

En contra, este tipo de firewalls pueden presentar vulnerabilidades que puedan ser aprovechadas por un atacante. Además, no suelen tener activadas sus capacidades de registro, dificultando al administrador la detección de un posible ataque.

Su capacidad de actuación puede deteriorarse si se utiliza un filtrado muy estricto, llegando a dificultar la gestión del firewall si el número de reglas es muy alto.

Un ejemplo de este tipo de firewalls es el módulo Netfilters/iptables del núcleo de Linux (García Alfaro, 2004).

2.4.1.2. Pasarelas a nivel de aplicación

Las pasarelas a nivel de aplicación (proxy), no encaminan paquetes a nivel de red, sino que trabajan a nivel de aplicación. Los usuarios de la red, se tendrán que conectar, en primer lugar, al servidor proxy asociado a la aplicación concreta.

Las pasarelas separan completamente la red interna de la externa a nivel de enlace. La pasarela ofrece servicios de nivel de aplicación frente a los que tiene que autenticarse el usuario que desea realizar una petición de conexión. Estas características hacen que las pasarelas a nivel de aplicación se presenten como alternativa que ofrece más seguridad respecto a los filtros de paquetes. En cambio, se introduce una penalización, al trabajar a un nivel de abstracción mayor, provocando que el rendimiento caiga drásticamente en momentos de alta carga de la red.

Las pasarelas ofrecen varias ventajas, entre ellas:

- Las pasarelas únicamente permiten el acceso a los servicios para los cuales existe un servidor proxy habilitado. Por tanto, si en nuestra red solo existe un proxy para el servicio HTTP, podemos estar seguros de que únicamente entrarán en la red interna paquetes de ese protocolo.
- Las pasarelas permiten la prohibición de diferentes sub-servicios dentro de un mismo servicio permitido. Por ejemplo permitir o rechazar determinados comandos de un protocolo.
- Los servidores intermediarios permiten aplicar filtros que discriminen en función de la dirección IP de la conexión, al igual que hacen los filtros de paquetes.

A pesar de la potencia que ofrecen las pasarelas, hay que tener en cuenta una gran desventaja que poseen, y es que hay que configurar un servidor proxy para cada servicio que se desea vigilar (García Alfaro, 2004).

2.4.1.3. Pasarela a nivel de circuito

Las pasarelas a nivel de circuito son un híbrido entre las plataformas a nivel de servicio y filtrado de paquetes. Son dispositivos en los que el usuario establece primero la conexión con el sistema de cortafuegos, y este a su vez, establece la conexión con el equipo de destino.

La diferencia con una pasarela tradicional reside en que una vez establecida la conexión, el firewall funciona como un filtro de paquetes. De manera que con la

conexión establecida, el firewall retransmite los paquetes entre ambos extremos sin inspeccionar su contenido.

La ventaja de este tipo de dispositivos es la velocidad a la que operan. Ya que no es necesario analizar todo el contenido de los paquetes transmitidos (García Alfaro, 2004).

2.4.2. Copias de seguridad (Backup)

Backup y recuperación es un tema que parece básico a primera vista, pero resulta confuso para una gran cantidad de gente. Los términos "*backup*" y "*archivo*" suelen usarse indistintamente para referirse a cualquier tipo de protección temporal de la información. Además, las empresas contribuyen a esta confusión al agrupar ambas funciones en un único grupo, enfatizando la parte de backup de datos, ofreciendo la visión de que son una única cosa.

La diferencia entre backups y archivos se vuelve especialmente confusa cuando los backups son almacenados durante largos periodos de tiempo, del orden de años. Estos backups pueden ser incorrectamente denominados "archivo" porque la información que contiene el backup puede ser la única copia de la información existente en un determinado instante temporal.

Los backups son copias instantáneas tomadas en un instante temporal concreto, almacenadas en formato determinado, y mantenidos durante un periodo de tiempo en el cuál son útiles, manteniendo cada copia de información de manera independiente a la primera de ellas (Nelson, 2011).

Consideramos que un *backup completo* es una instantánea en un momento determinado de toda la información que se desea respaldar. Tomando esta instantánea como base, se puede transformar mediante los siguientes algoritmos, para alcanzar una situación de compromiso con los recursos de los que se dispone.

2.4.2.1. Backup diferencial

El backup diferencial se realiza comparando dos ficheros de backup consecutivos, extrayendo en el proceso un tercer fichero que contiene las diferencias entre los dos ficheros de backup. De tal manera que a partir del fichero de diferencia, y uno de los backups originales, sería posible poder reconstruir el segundo fichero de backup original.

Así, si los ficheros obtenidos mediante dos backups consecutivos son similares, el fichero que describe la diferencia entre ambos, debería de ser menor que cualquiera de los dos. Aquí es donde se obtiene la mejora: se almacena únicamente un backup "**padre**" con sucesivos backups "**hijos**" calculados como la diferencia del backup **padre** con el resto de los backups posteriores.

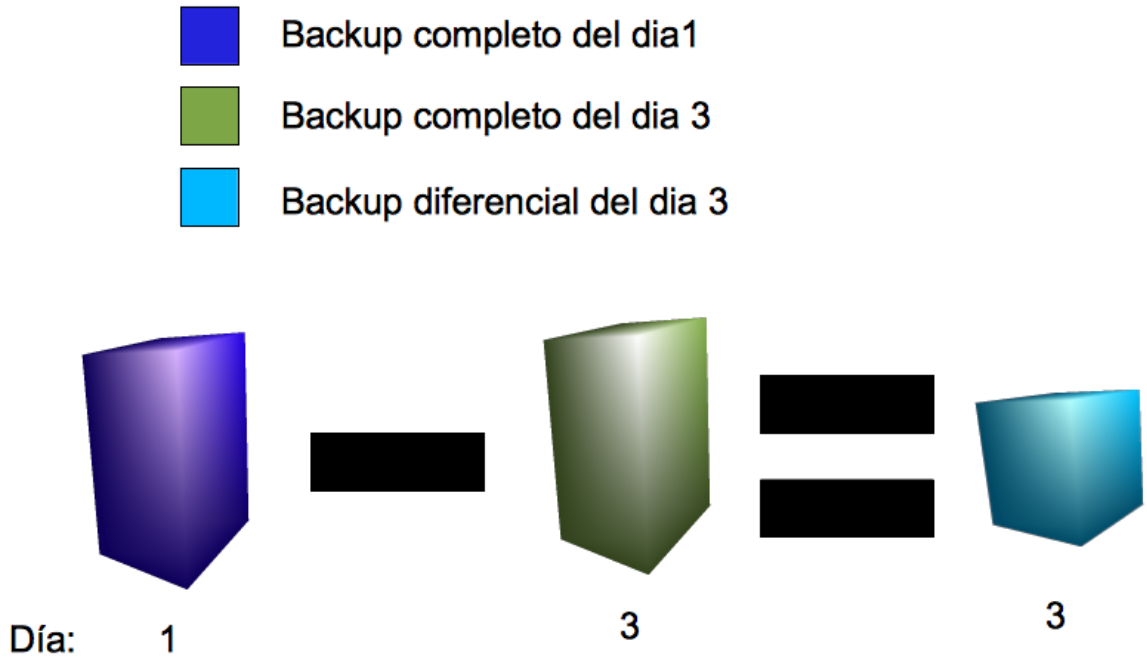

Ilustración 7 Creación de un backup diferencial

En la figura *Ilustración 7 Creación de un backup diferencial* se muestra esquemáticamente como se realiza el proceso de backup, representando mediante el símbolo "-" el proceso de extraer las diferencias del backup del día 3 respecto al backup del día 1.

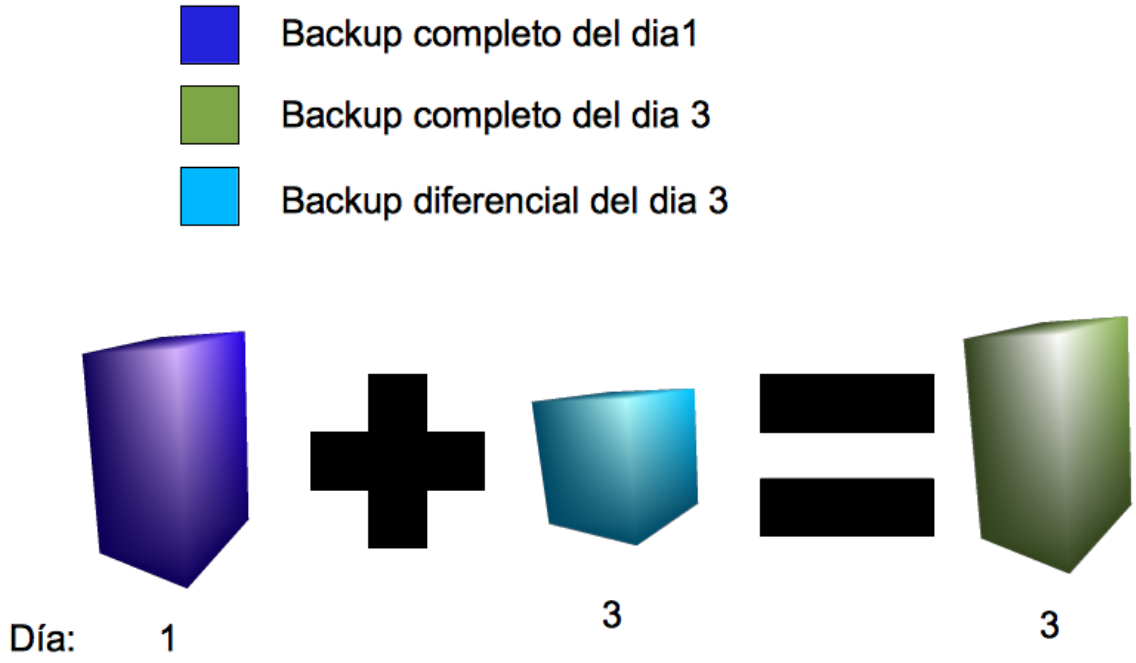

Ilustración 8 Restauración de un backup diferencial

La *Ilustración 8 Restauración de un backup diferencial* resume esquemáticamente el proceso de restauración del backup del día 3 original. Como se puede observar, sólo se almacena el fichero con las diferencias ente el backup del día número 3 y el del día 1. Mediante el símbolo "+" se indica el proceso de modificar el elemento a la izquierda con los cambios que especifica el elemento a la derecha.

Una vez definidas las bases del backup diferencial se pasa a explicar en qué consiste el algoritmo. Se define el término "**ciclo**", que es el número de backups (incluyendo el de referencia) que se realizan antes de escoger un nuevo backup completo de diferencia. Como ejemplo se supone el caso más general (que es el que se desea implementar) de realizar un backup por día, de tal manera que se pueda identificar un backup mediante el número de día en que se realizó. A continuación se describe el proceso para realizar dos ciclos completos de cuatro elementos cada uno.

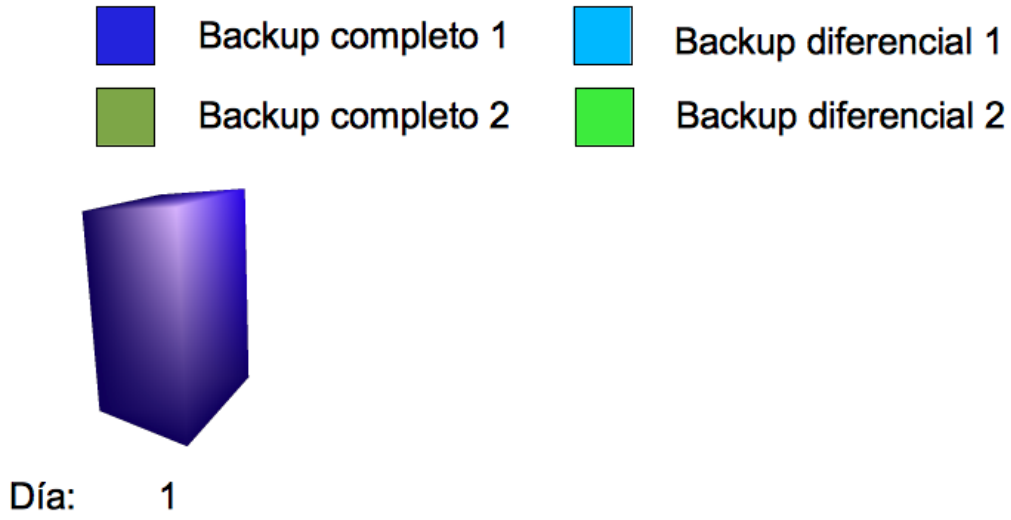

Ilustración 9 Backup diferencial: día 1

El algoritmo de backup en el día 1 se limita únicamente a crear un backup normal mediante el uso de una herramienta que realice una copia binaria del disco duro del equipo que se quiere salvar, y se almacena en un directorio sin realizar ninguna modificación.

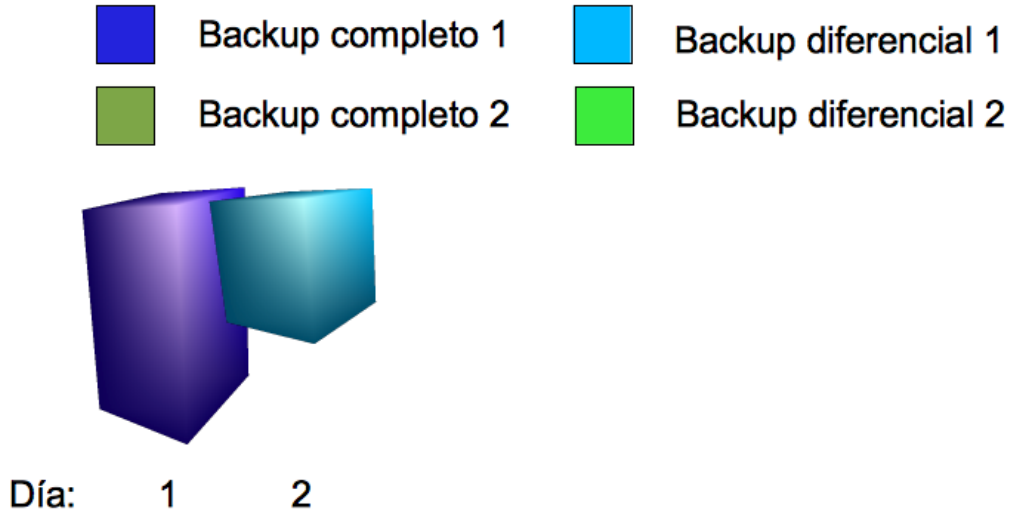

Ilustración 10 Backup diferencial: día 2

Dentro del primer ciclo, se procede a realizar el primer backup diferencial en el día 2. Para llevarlo a cabo, primero se realiza un backup completo normal del día 2 almacenándolo en un directorio temporal. Una vez finalizado el volcado, se recupera el backup de referencia del ciclo, que en este caso será el backup del día 1, y se calcula la diferencia que existe entre ambos backups. Se almacena el fichero de diferencia etiquetándolo como backup del día 2 en el mismo directorio que el

backup del día 1 y, por último, se elimina el backup completo del día 2 del fichero temporal.

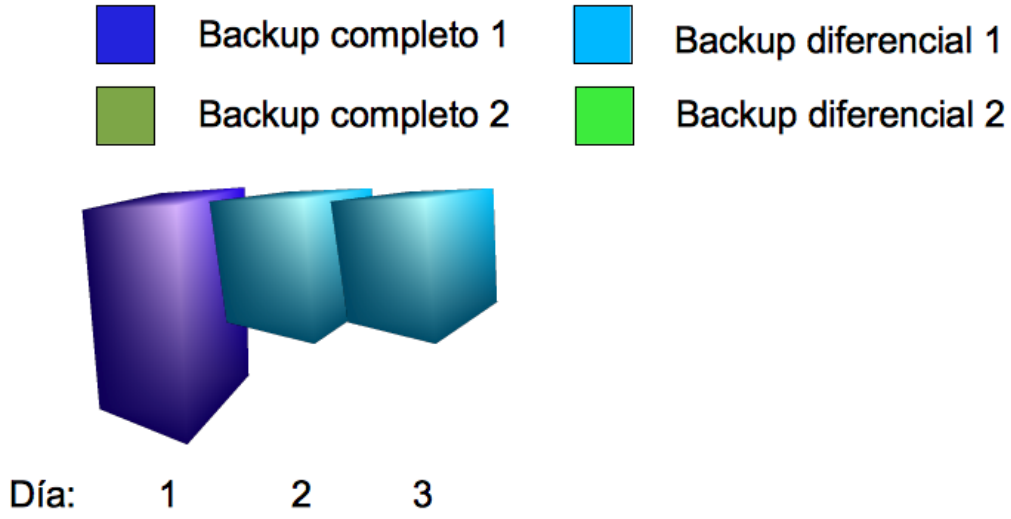

Ilustración 11 Backup diferencial: día 3

El proceso en el día 3 y 4 se realiza de forma análoga a como se realizó en el día 2. Es decir, se crea un backup completo del día en un directorio temporal, se compara con el backup de referencia del día 1 y se almacena su diferencia como backup del día correspondiente, eliminando el backup completo temporal.

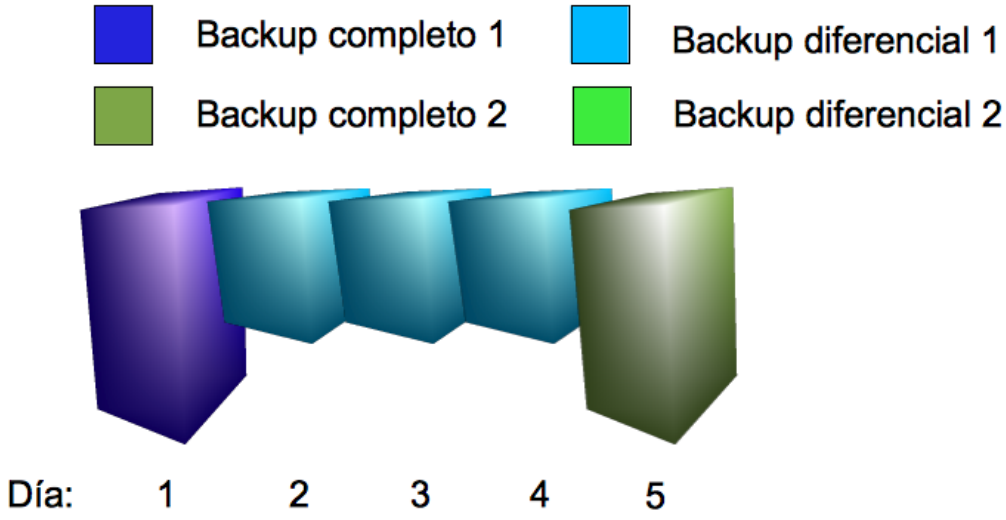

Ilustración 12 Backup diferencial: día 5

Puesto que los ciclos eran de 4 días, el día 5 marca el inicio de un nuevo ciclo de backup. Como se comienza un ciclo nuevo, se almacena el backup del día 5 sin modificación ninguna en el directorio junto a los otros backups. Ahora en los pasos siguientes será el backup del día 5 el backup de referencia.

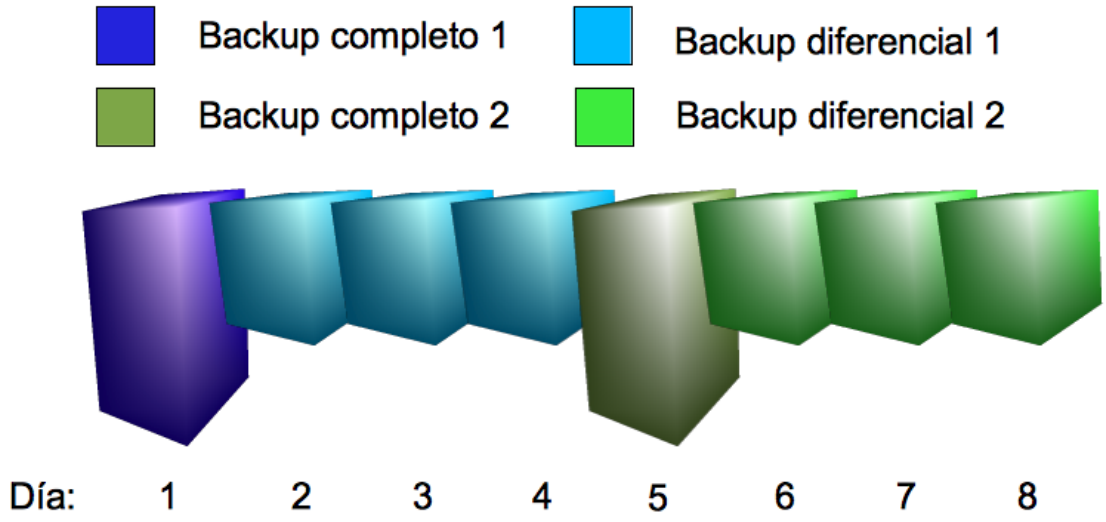

Ilustración 13 Backup diferencial: día 8

Hasta el día 8, en el cual termina el segundo ciclo, el backup diferencial se continúa haciendo de la misma forma creando tres nuevos ficheros de backup que contienen sus diferencias respecto al backup del día 5.

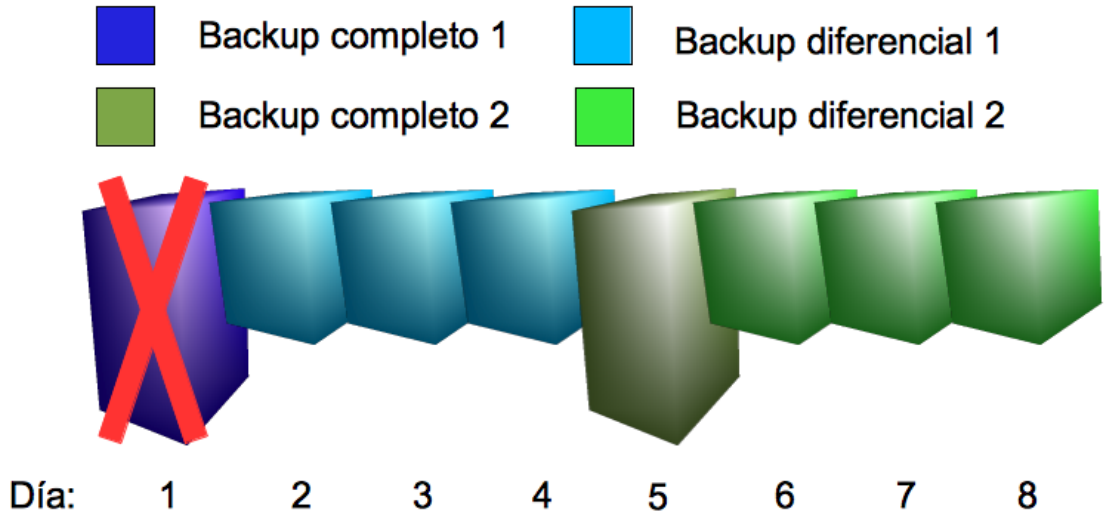

Ilustración 14 Backup diferencial: día 9, paso 1

Al mantenerse los backups de dos ciclos, al comienzo del backup del día 9, que es cuando se comienza el tercer ciclo, hay que eliminar el backup más antiguo almacenado para hacer sitio al nuevo backup completo del día 9.

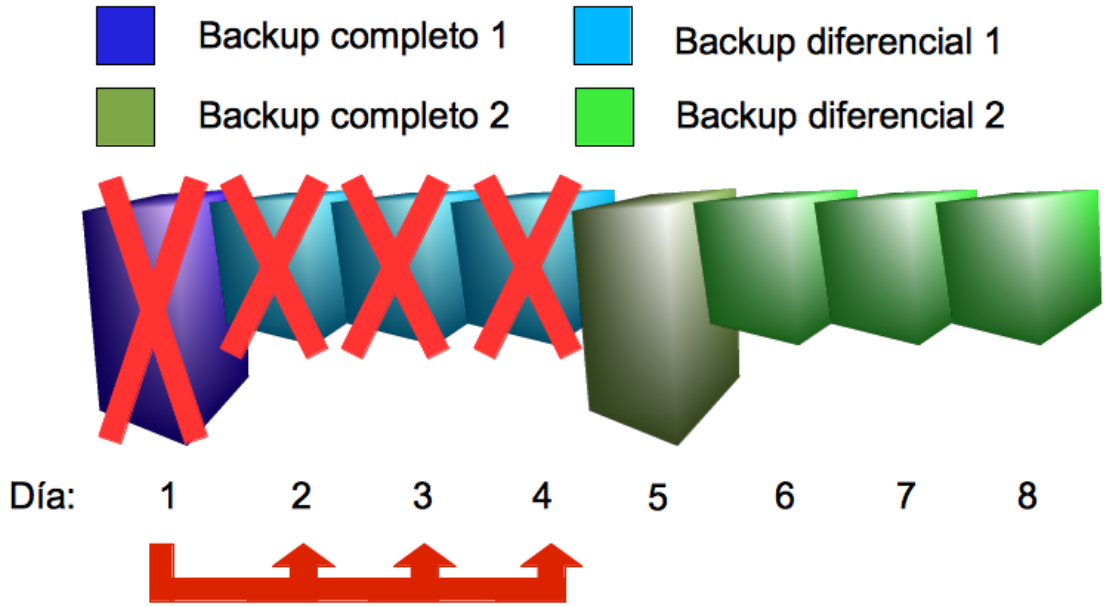

Ilustración 15 Backup diferencial: día 9, paso 2

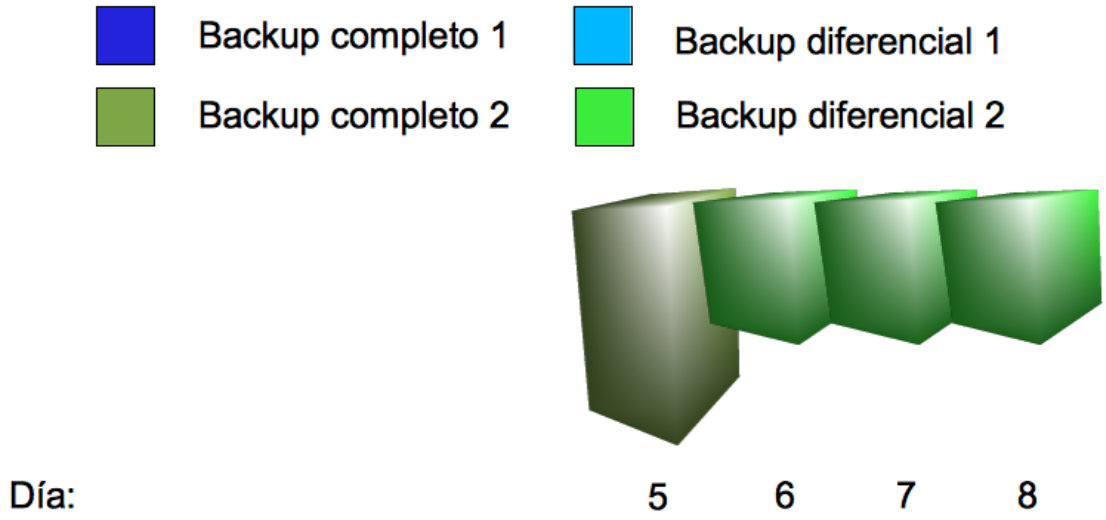

Ilustración 16 Backup diferencial: día 9, paso 3

El backup más antiguo en este caso es el del día 1, pero al eliminar éste backup, todos los backups diferenciales creados en ese ciclo dejarían de poder ser restaurados, al no haber un backup de referencia. Esos backup diferenciales ya no tienen ninguna función, y son eliminados para ahorrar espacio de almacenamiento.

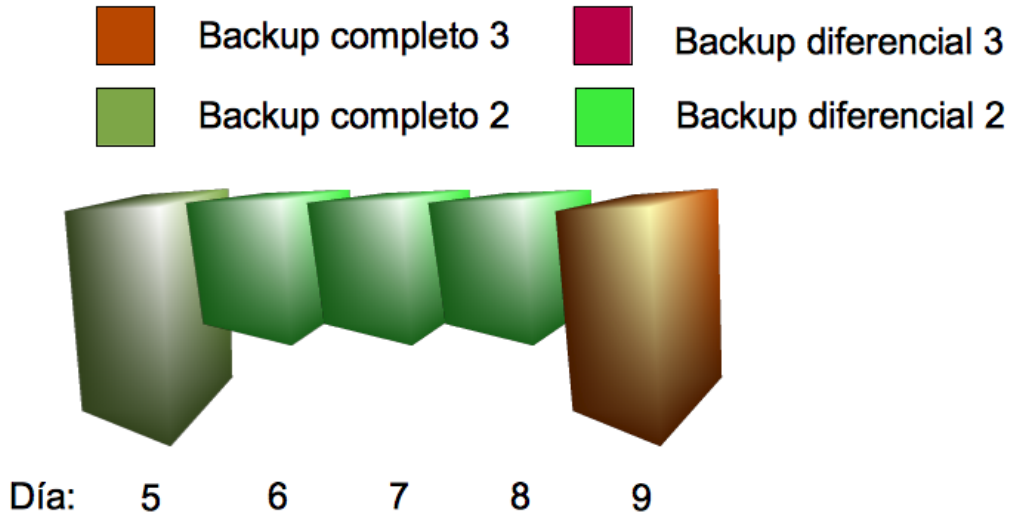

Ilustración 17 Backup diferencial: día 9, paso 4

Una vez eliminados los ficheros de backup se continúa con la generación del backup completo del día 9 de la forma habitual y se almacena en el directorio junto el resto de backups pasando a ser el nuevo backup de referencia.

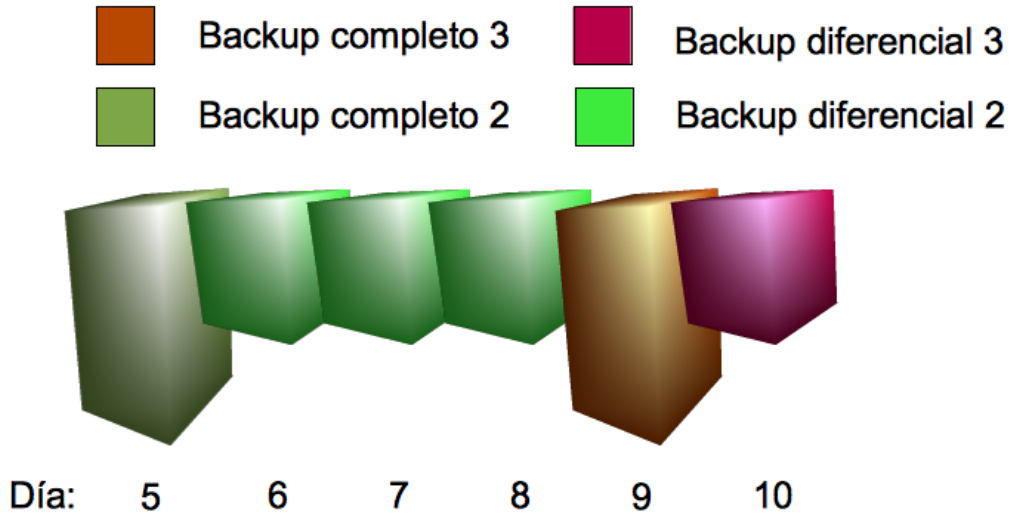

Ilustración 18 Backup diferencial: día 10

Finalmente, el backup del día 10 se realiza de forma diferencial, sin necesidad de eliminar el backup más antiguo, dado que en el paso anterior se eliminaron más de un backup y se tienen almacenados menos de $Nº\ de\ ciclos \times Tamaño\ de\ ciclo$. Por lo tanto sólo se pueden eliminar backups cuando se almacena una copia de seguridad o completa (de referencia).

2.4.2.2. Backup incremental

El backup diferencial presenta problemas cuando se producen grandes cambios entre imágenes. Los cambios se propagarán en los backups diferenciales que queden hasta terminar el ciclo. Esto implica almacenar información redundante que conlleva la pérdida del ahorro de espacio del backup diferencial, manteniendo todas sus desventajas: perder todos los backups de un ciclo cuando se comienza un ciclo nuevo y todas las operaciones que conlleva calcular las diferencias.

El backup incremental subsana este problema almacenando únicamente los cambios que se han producido ese día. A cambio, incrementa el número de cálculos que se tienen que realizar para generar una copia de seguridad, con el consiguiente aumento de tiempo para la generación del backup.

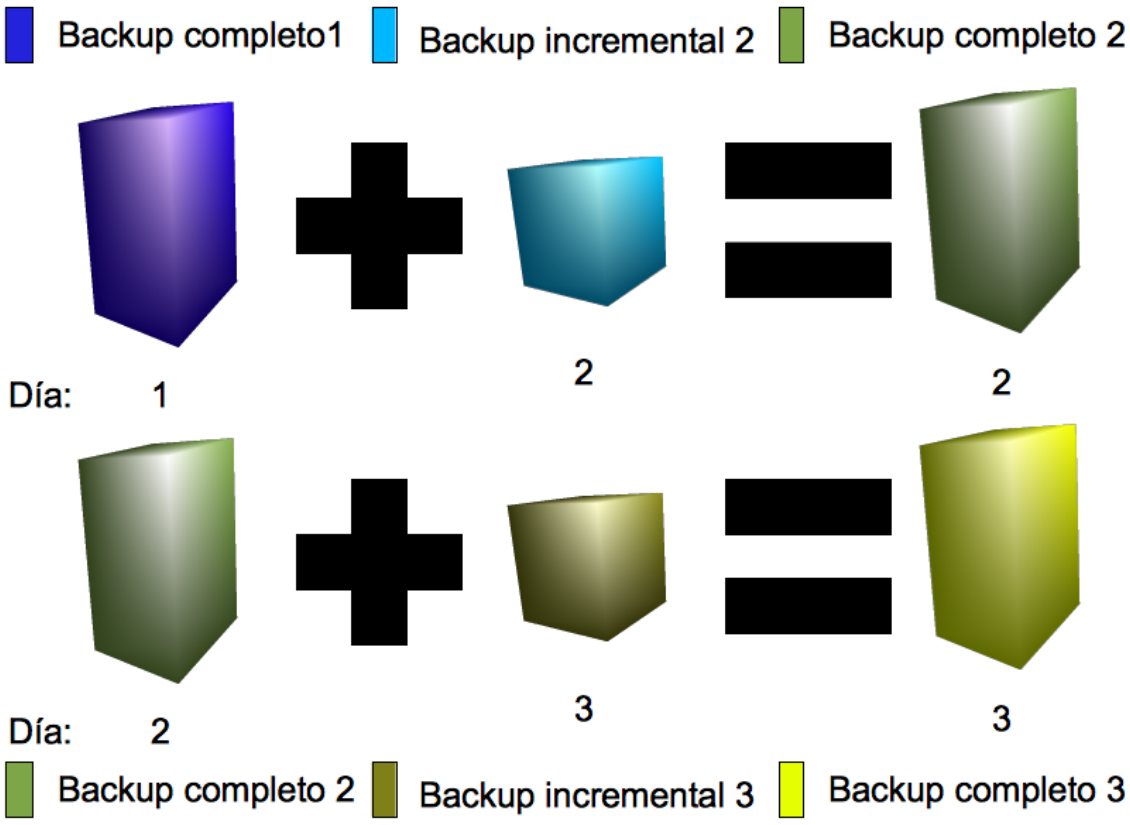

Ilustración 19 Restauración de un backup incremental

El algoritmo de backup incremental se diferencia del diferencial en la operación que se realiza para crear un backup y en restaurarlo. En la *Ilustración 19 Restauración de un backup incremental* se indica el proceso para reconstruir el backup completo del día 3 a partir de todos los fragmentos de los backups anteriores hasta su backup de referencia. De este modo, para obtener el backup del día 3 hay que aplicar el fichero con las diferencias del día 3 respecto a al backup completo del día 2. Pero el backup completo del día 2 no está almacenado, solo se tiene la diferencia del día 2 respecto al backup completo del día 1, que sí está almacenado al ser el backup de diferencia. Por lo tanto, para restaurar un backup incremental hay que restaurar cronológicamente todos los backups que haya almacenados desde el backup de referencia, que puede ser una operación costosa.

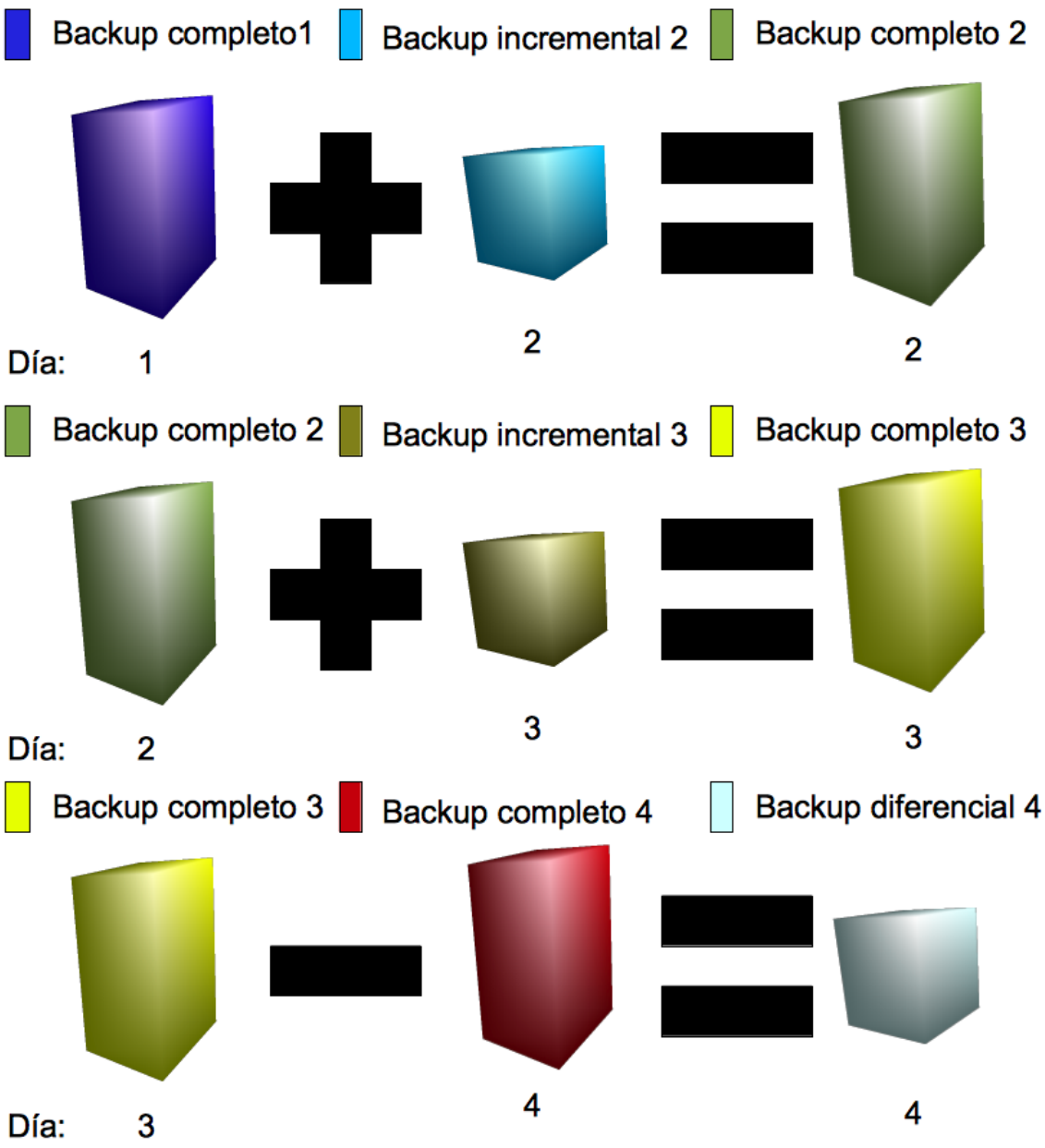

Ilustración 20 Creación de un backup incremental

La creación de los backups incrementales de los días 1 y 2 se realizan de forma idéntica a como se explicó anteriormente para el backup diferencial en las imágenes *Ilustración 9 Backup diferencial: día 1* e *Ilustración 10 Backup diferencial: día 2*. La idea de realizar un backup incremental es conceptualmente muy sencilla, únicamente basta con calcular la diferencia entre el backup del día actual con la del día anterior. El problema surge cuando no tenemos un "*backup del día anterior*" con el que comparar directamente, teniendo que calcular previamente cuál es el backup completo del día anterior para poder extraer los cambios. En *Ilustración 20 Creación de un backup incremental* se muestra un ejemplo del peor caso a la hora de generar un backup, suponiendo un tamaño de

ciclo de 4 backups. En este caso, para crear el backup incremental del día 4 es necesario tener el backup completo del día 3 para calcular su diferencia, puesto que el backup del día 3 es incremental, es necesario el backup completo del día 2 para poder calcularlo, lo que implica calcular el backup completo del día 2 a partir del backup de referencia, que es el del día 1.

El concepto de ciclos desarrollado anteriormente se sigue aplicando aquí, generando cada cierto número de backups incrementales un nuevo backup completo que sirva como nueva referencia. De esta manera se acota el número de operaciones máximas a realizar para completar un backup. La desventaja es que se hereda el inconveniente de que cuando se elimina un backup de referencia, por quedar obsoleto, se seguirán borrando todos los backups hijos de ese mismo ciclo.

2.4.2.3. Backup diferencial exhaustivo

Este nombre es el que se le ha dado al algoritmo derivado del backup diferencial que subsana el problema de perder un ciclo completo de backups cada vez que se elimina un backup de referencia, teniendo siempre el máximo posible de puntos de restauración de backup.

El funcionamiento es idéntico al del backup diferencial, tanto en las operaciones de restauración y creación del backup, como en el desarrollo basado en backups de referencia y un conjunto de backups parciales dependientes de él dentro de su mismo ciclo. La principal diferencia radica en la operación de eliminar backups obsoletos. En primer lugar, siempre que se crea un nuevo backup hay que eliminar el backup más antiguo, ya sea diferencial o completo el nuevo backup generado y, por ende, el backup eliminado. La segunda particularidad del borrado surge en el momento de eliminar un backup diferencial, al desaparecer su backup de referencia, es decir, no le tocaba ser borrado. En este caso, antes de eliminar el backup de referencia, se generan con él todos los backups completos de su ciclo en un directorio temporal, momento en el que ya se puede borrar el backup de referencia al no ser necesario. A continuación, para cada backup completo generado de este modo del directorio temporal, se compara con el backup de referencia más cercano (el primer backup del siguiente ciclo) generando $Tamaño\ de\ ciclo - 1$ nuevos backups diferenciales. Una vez generados los backups diferenciales, se puede eliminar el directorio temporal con todos los backups completos que contenía.

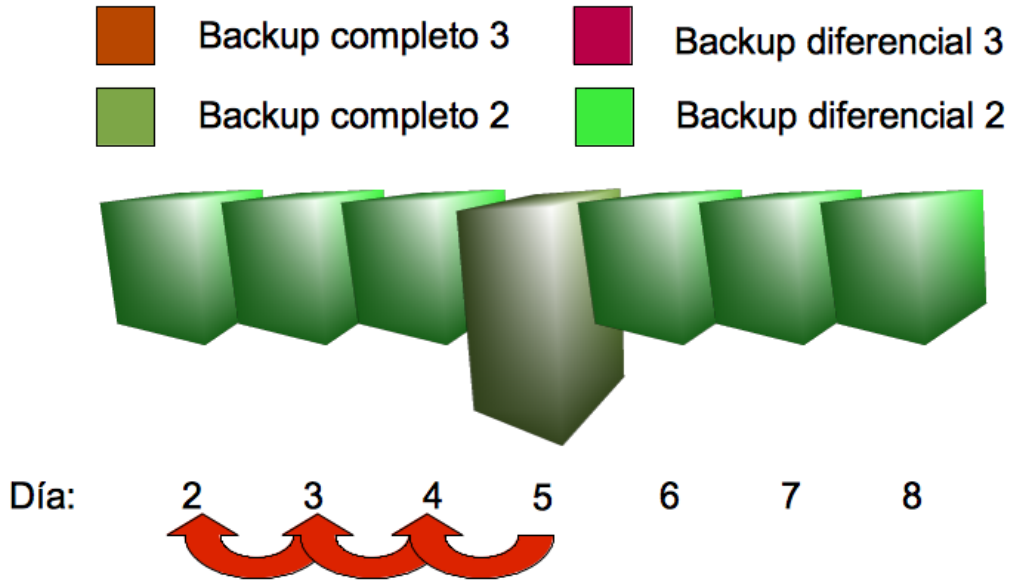

Ilustración 21 Backup diferencial exhaustivo: día 9, paso 4

Supongamos que se ha seguido el proceso de creación del backup diferencial hasta llegar al paso de la *Ilustración 15 Backup diferencial: día 9, paso 2*, en ese punto, antes de eliminar los backups de los días 1, 2 ,3 y 4, se restauran a partir del backup del día 1 los backups de los días 2, 3 y 4. A continuación, teniendo como referencia el backup completo del día 5, se compara por separado con los del día 2, 3 y 4 calculando su backup diferencial y almacenándolos junto al resto de backups.

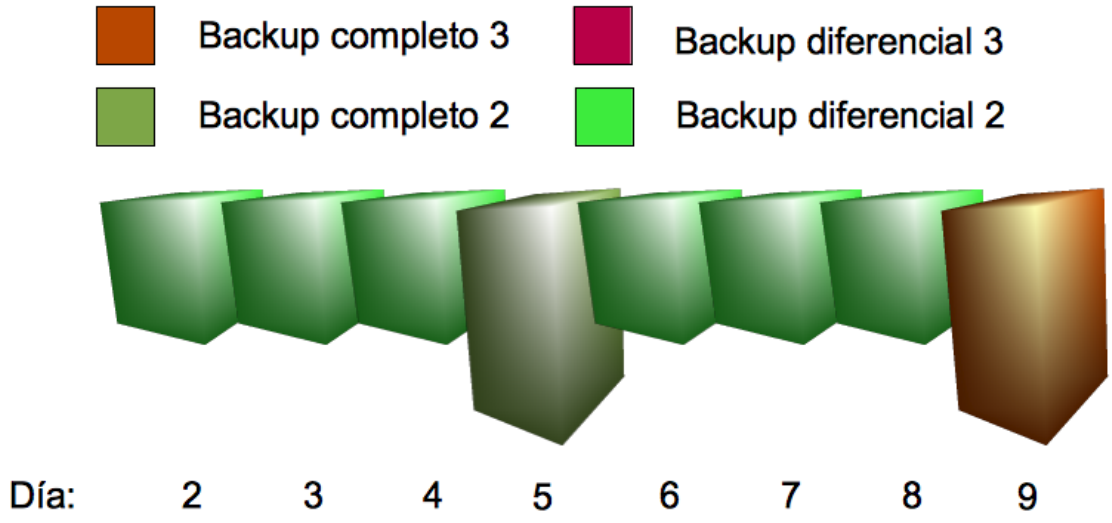

Ilustración 22 Backup diferencial exhaustivo: día 9, paso 5

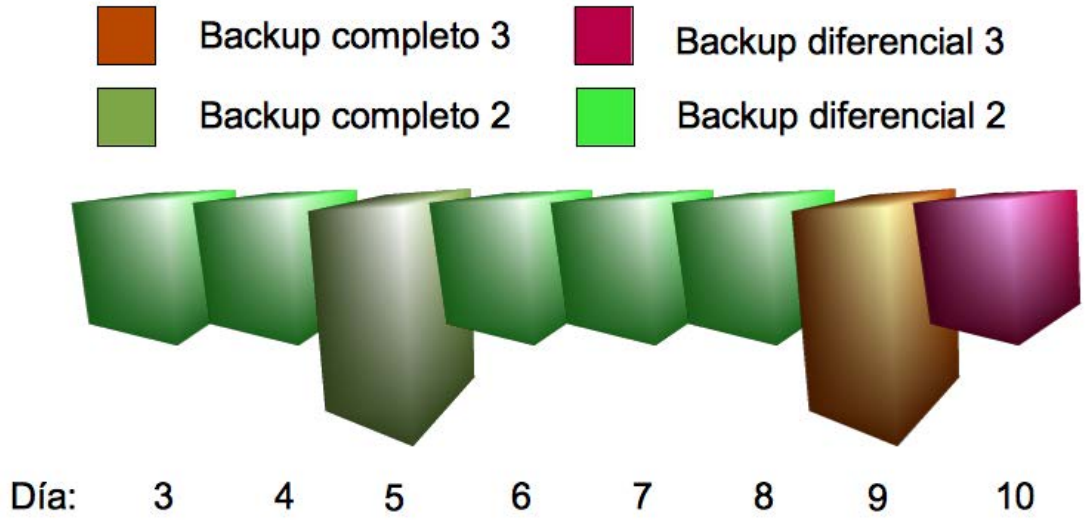

Ilustración 23 Backup diferencial exhaustivo: día 10

El procedimiento continúa de la forma habitual generando el backup completo de referencia del día 9, una vez se ha terminado el proceso de eliminar el backup de referencia del día 1. El procedimiento varía a partir del día 10, en el cual se crea un nuevo backup diferencial tomando como referencia el backup completo del día 9, pero antes de crearlo, hay que eliminar el backup diferencial del día 2, el cual se elimina sin ninguna operación adicional a no haber backups dependiente de él.

2.4.2.4. Backup normal

El backup normal se limita simplemente a almacenar en el directorio de backups el fichero que se genera al extraer una copia binaria del disco duro de la máquina a salvar, sin realizar ninguna modificación. Teniendo de esta manera una copia completa todos los días, es decir, sin perder ciclos completos al eliminar un backup de referencia y sin tener que hacer operaciones costosas en tiempo para eliminar la redundancia. En contra, no se obtiene ningún ahorro de espacio de almacenamiento, si se da la situación en que el espacio de almacenamiento es fijo y limitado, el número de puntos de restauración disponibles con éste método es inferior al de las otras alternativas.

2.4.2.5. Backup por ficheros

La última alternativa para realizar el backup utiliza un enfoque distinto al de las otras cuatro. Mientras que las primeras opciones de backup trataban con los

discos completos en formato binarios, éste método intenta hacer la copia de seguridad a un nivel de abstracción mayor, copiando los ficheros que contiene ese disco a salvar. Esto permite una mayor eficiencia en el backup ya que se pueden utilizar herramienta de Unix como *tar*, preparadas para realizar backups incluso de forma incremental, sin necesidad de tener que desarrollar ninguna aplicación que implemente el algoritmo.

2.5. Monitorización

La monitorización es el punto principal del proyecto, no por que posea una importancia mayor, sino porque es donde se ha invertido un mayor esfuerzo en su desarrollo y documentación. Por ello en este punto, se mostrará una serie de herramientas existentes en el mercado (todas ellas basadas en software libre) que permiten realizar la función de monitorización, a tener en cuenta durante el desarrollo del proyecto. Además en primer lugar se introducirá en que consiste y que ofrece la monitorización.

2.5.1. ¿Qué es la monitorización?

Son varias las razones para monitorizar una red, entre ellas, la más común es poder conocer cuando está fallando algo. Además de permitir detectar los fallos, monitorizar también permite predecir determinadas situaciones y evitar algunas desagradables. Por ejemplo, si conocemos el dato sobre que la capacidad de un disco está a punto de agotarse, podemos reaccionar a tiempo antes de que llegue a llenarse (con posibles pérdidas de información), liberando espacio en el disco o ampliando su capacidad.

Otra gran ventaja que aporta la monitorización de la red es que facilita los planes de ampliación. Por ejemplo, una tendencia que busca en el pasado y proporciona datos de históricos nos informará sobre qué emails se ven incrementados en un 10% mensual, ayudándonos a decidir si en x meses será necesario un nuevo servidor de correo o no.

Por otro lado tenemos la denominada monitorización de estados. Con este tipo de monitorización se responde principalmente a preguntas del tipo “¿está activo este proceso?”. Esta funcionalidad es la base de la mayoría de los sistemas de gestión de redes.

La segunda función más popular, es la monitorización de rendimientos, o tendencias. Un monitor de rendimientos nos indica, por ejemplo, que durante los últimos meses la carga de CPU ha rondado en torno al 20%, y recientemente está teniendo una carga del 80%. Esta información es especialmente importante cuando se desea analizar elementos como el ancho de banda.

Por último, tenemos la monitorización de registros o logs. Prácticamente cualquier aplicación genera logs en los que registra información sobre lo que está sucediendo y que puede estar yendo mal.

El conjunto de todas estas herramientas mencionadas es lo que se conoce como **sistema de monitorización de redes** (Nalley, 2009).

2.5.2. Nagios

Ilustración 24 Logotipo de Nagios

Nagios es una herramienta para la monitorización de sistemas que permite a las organizaciones identificar y resolver problemas en su infraestructura informática antes de que puedan afectar a procesos críticos del negocio (Nagios Enterprises, 2011).

Originariamente fue lanzado en 1999 bajo el nombre de "NetSaint" que tuvo que ser reemplazo al poco tiempo debido a problemas de derechos, al ser una marca que ya existía por el nombre de "Nagios". Nagios es de código abierto y licenciado bajo la licencia GNU GPLv2.

Nagios no realiza las mediciones de ningún equipo o servicio, sino que son las extensiones las encargadas de realizar esas mediciones. Convirtiéndolo en una solución muy modular y flexible para la medición de rendimiento.

Los objetos monitorizados por Nagios se dividen en dos categorías: equipos y servicios. Con equipos se refiere a máquinas físicas como pueden ser servidores, enrutadores, estaciones de trabajo, impresoras, etc. Mientras que como servicios interpreta funcionalidades particulares, como un servicio web que puedes ser definido para ser monitorizado. Cada servicio está asociado con un equipo en el que se está ejecutando. Además tanto equipos como servicios se pueden agrupar en grupos de servicio o equipo.

Nagios cuenta con dos principales virtudes, la primera de ellas es que en vez de monitorizar valores numéricos, utiliza cuatro valores nominales para describir el estado de un elemento: "OK", "WARNING", "CRITICAL" y "UNKNOWN". Mostrar únicamente valore abstractos de monitorización permite a los administradores olvidarse de tener que consultar las mediciones y simplemente decidir cuáles son los valores que consideren como los límites de lo que es una advertencia o una situación crítica. Tener un límite estricto es mucho más sencillo a la hora de vigilar un elemento monitorizado ya que puedes ver el momento exacto en el que se produce una advertencia y poder corregirla antes de sobrepasar el límite indicado como crítico.

Otra ventaja es que los informes muestran todos los servicios que están funcionando, ya se encuentren en estado de advertencia o crítico, ofreciendo una visión general de la infraestructura monitorizada, permitiendo localizar anomalías rápidamente y priorizar cuales son los problemas que se van a subsanar antes y cuáles pueden ser dejados para más adelante. Los informes se pueden visualizar alternativamente por grupos de equipos o grupos de servicios.

Como se mencionó anteriormente, Nagios en sí no se encarga de la monitorización de elementos concretos, sino que es una tarea que delega en las extensiones instaladas, de modo que hay que instalar las extensiones adecuadas en función de lo que se quiera monitorizar, o programarlas en algún lenguaje de programación. El funcionamiento es el siguiente: Nagios envía a las extensiones cuales son los elementos que se deben monitorizar, así como los valores que tienen los límites que definen los estados ("OK", "WARNING", "CRITICAL" y "UNKNOWN"), la extensión por su parte únicamente devuelve cuál de los cuatro estados es el que presenta el elemento medido y opcionalmente puede incluir un campo de descripción que puede indicar al administrador datos más concretos (Kocjan, 2009).

Las características principales que ofrece Nagios son:

- La gran virtud de Nagios es su flexibilidad, permite configurar la monitorización de la infraestructura mediante mecanismos que reaccionan automáticamente ante los problemas, y posee un potente sistema de notificaciones.
- Permite definir dependencias mediante equipos de tal manera que un fallo en un elemento, desactiva la monitorización en todos los elementos que dependen de él. Si suponemos que tenemos una red con varios equipos monitorizados, la cual es accesible por un único punto de acceso que es el enrutador (también monitorizado). En caso de que por cualquier motivo dejase de funcionar el enrutador, todos los equipos de su red dejarían de estar accesibles, provocando que saltasen todas las alertas de los equipos

aún sin ser ciertas. Mediante esta funcionalidad se puede decir que todos los equipos de la red dependen del enrutador, así ante cualquier problema en el enrutador, solo se notificara el problema que tiene el enrutador, ignorando el resto de equipos de la red.

- Permite la definición de dependencias entre servicios, de manera que un determinado servicio depende de otro servicio ya sea del mismo equipo o de otro diferente. Si un servicio del que dependen otros deja de funcionar, se dejarán de comprobar todos los servicios dependientes.
- Nagios cuenta con un sistema de definición de macros. Las macros son variables que pueden ser utilizadas en cualquier definición de medición tomando un valor diferente en función de su contexto. Un ejemplo de macro podría ser IP refiriéndose a la dirección IP de un equipo, dependiendo del equipo en el que se utiliza la macro, dará un resultado u otro dependiendo de cuál sea la dirección IP de ese equipo.
- Ofrece mecanismos para definir periodos de tiempo de cese de servicios planificados. Permite planificar cuando un determinado equipo o servicio no estará disponible, evitando las alertas que pueda originar y no solo eso, sino programar notificaciones que alerten a los usuarios informándoles de que determinado servicio estará temporalmente inaccesible.

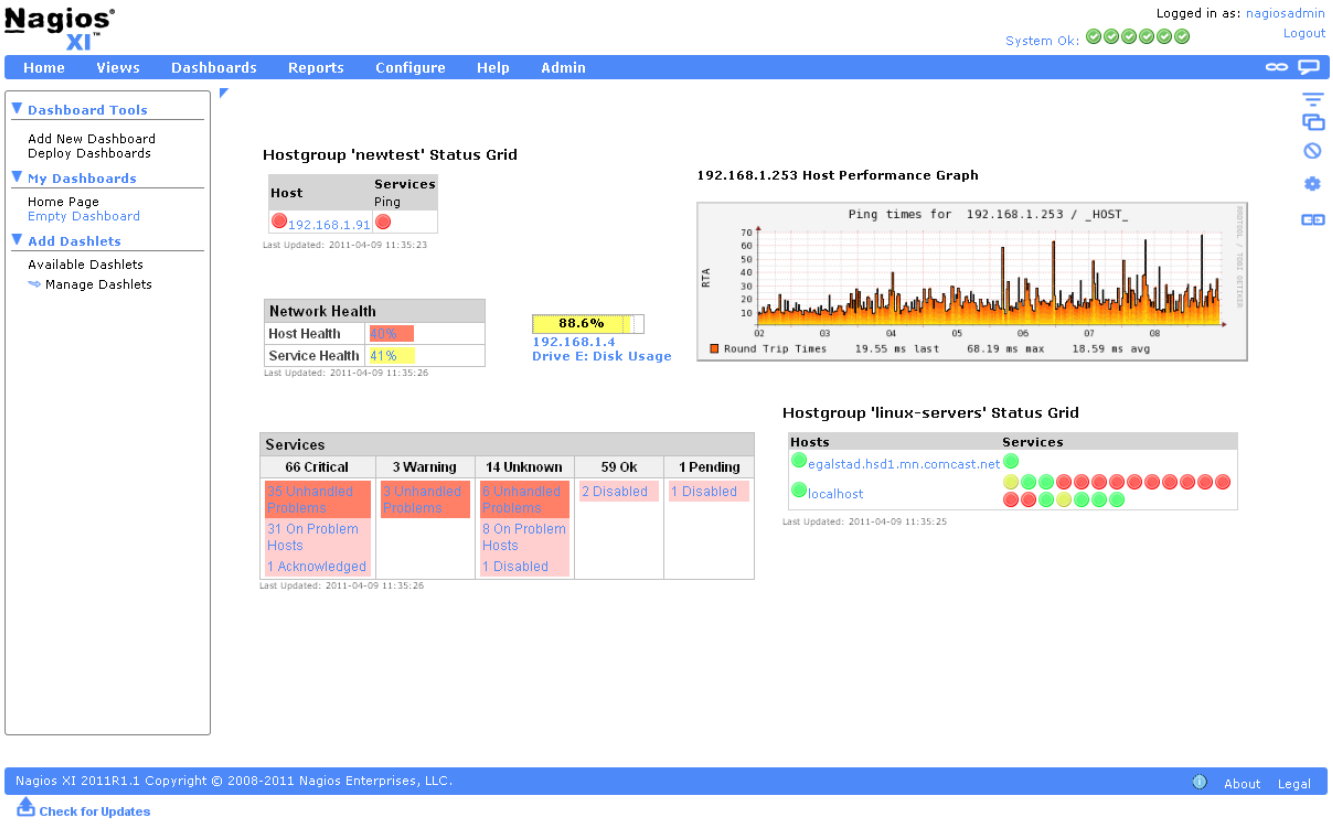

Ilustración 25 Pantalla "Dashboad" en Nagios

En la *Ilustración 25 Pantalla "Dashboad" en Nagios* se ve un ejemplo de la interfaz gráfica que presenta Nagios.

2.5.3. Pandora FMS (Flexible Monitoring System)

Ilustración 26 Logotipo de Pandora FMS

PandoraFMS es una aplicación de monitorización de todo tipo de sistemas y aplicaciones, permitiendo conocer el estado de cualquier elemento de un sistema de negocio. Entre los elementos monitorizados por Pandora FMS están el software, aplicaciones, hardware o sistema operativo, pudiendo detectar si se ha caído una interfaz de red o notificar mediante un SMS valores bursátiles.

Pandora FMS no se limita únicamente a medir si un parámetro está bien o mal, sí que permite cuantificar el estado (bien, mal o valores intermedios) o almacenar un valor de cualquier tipo durante meses. También permite medir rendimientos o comparar valores entre distintos sistemas para establecer alertas si se exceden ciertos umbrales.

La aplicación funciona sobre una base de datos, que le permite generar informes, estadísticas o niveles de adecuación de servicio. Permite hacer mediciones sobre cualquier cosa que sea capaz de generar datos, ya sean sistemas operativos, servidores o cualquier tipo de sistema hardware como cortafuegos, agentes (proxies), bases de datos, servidores web, VPN, encaminadores (routers), conmutadores (switches), procesos, servicios, acceso remoto a servidores etc. En una arquitectura abierta y distribuida.

Existen agentes para poder monitorizar todo tipo de sistemas, desde sistemas Windows (2000, XP, 2003, 2008, Vista o 7) hasta sistemas Unix como Linux, Mac, Solaris, HP-UX, BSD, AIX, IPSO y OpenWRT. Aunque no se limita únicamente a la monitorización mediante agentes, sino que permite monitorización además mediante el protocolo SNMP o pruebas de red (TCP, ICMP) permitiendo la monitorización de cualquier hardware de red con conectividad TCP/IP, como balanceadores de carga, impresoras, etc. Concretamente Pandora FMS puede monitorizar cualquier elemento

que devuelva un valor tras ejecutar un comando, así como cualquier valor almacenado en un registro de texto, sistema o fichero.

Pandora FMS está publicado bajo la licencia GNU GPLv2 que cuenta con una comunidad ,principalmente localizada en el foro, que responde a las preguntas de los usuarios, se informa sobre errores detectados, y se da asesoramiento respecto a integración con programas y dispositivos de terceros. No obstante, también se comercializa una versión Pandora FMS Enterprise con una licencia comercial que prohíbe su distribución aunque permite la modificación del código. El código de esta versión no está restringido por la licencia GPL al utilizar código escrito desde cero en vez de utilizar el de la versión libre de Pandora FMS. Esta versión Enterprise cuenta con soporte profesional, actualizaciones y mantenimiento automático a través del sistema Open Update Manager. Aparte de incluir las siguientes características no presentes en la versión libre:

- Open Update Manager conectado a Ártica: que aporta actualizaciones de software y acceso a bibliotecas del fabricante.
- Programador de informes: permite enviar los informes que genera Pandora FMS por correo electrónico. El envío de informes se puede programar con la frecuencia con la que se desea que se generen.
- Dashboard: muestra una pantalla de inicio con un resumen de las mediciones, la pantalla es configurable indicando lo que se quiere mostrar o añadir pestañas.
- Configuración remota de agentes: permite la personalización de los módulos que utilizan los agentes o configuración de los mismos como puede ser la dirección IP desde la consola Web.
- Gestión de políticas de monitorización: permite realizar agrupaciones de elementos de monitorización (alertas y módulos) llamadas políticas, que pueden ser aplicados directamente sobre un agente grupo de agentes. Cuando se modifica una política los cambios se aplican a todos los agentes suscritos a esa política, permitiendo la homogenización de las políticas de monitorización en un gran número de sistemas de forma cómoda y fácil.
- Servidor de exportación: aporta funcionalidad de escalado e datos, haciendo posible distribuir la infraestructura de monitorización entre varias instalaciones de Pandora FMS. El sistema distribuido está centralizado por una instalación de Pandora FMS “central” que es la que provee los datos al resto de instalaciones.

- Servidor de inventario: permite obtener de forma remota información sobre el inventario de los agentes (software instalado, parches, dispositivos hardware).
- Monitorización WEB transaccional: permite hacer comprobaciones complejas en sitios web que pueden involucrar navegación, rellenar formularios o uso de credenciales. Permite tanto medir los tiempos de carga como validar que realiza las tareas.
- Metaconsola: gestiona instalaciones independientes de Pandora FMS, incrementando la escalabilidad.
- ACL Enterprise: permite definir al administrador que secciones puede visualizar el usuario y cuáles no.
- Monitorización de servicios: Permite la monitorización de grupos de elementos (servicios) basada en pesos, de una forma mucho más flexible y con márgenes definidos por el usuario.

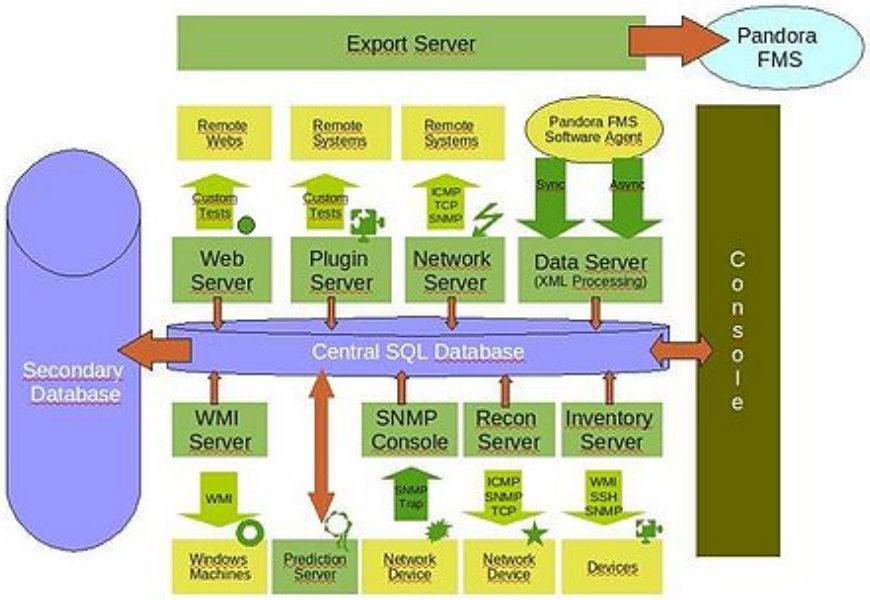

Ilustración 27 Arquitectura de Pandora FMS

Pandora FMS es un software modular, siendo el centro del sistema la base de datos MySQL (solo soporta la base de datos MySQL).Todos los componentes de Pandora FMS son replicables, y cuenta con diversos elementos y servidores. Los servidores son los encargados de recolectar y procesar los datos, para más adelante insertarlos en la base de datos. A los servidores les envían los datos los agentes que son aplicaciones que se ejecutan en los equipos monitorizados. Finalmente el último

componente es la consola, cuya función es mostrar al usuario los datos almacenados (openIdeas.info, 2011).

2.5.4. Zabbix

Ilustración 28 Logotipo de Zabbix

Zabbix es un software que monitoriza un gran número de parámetros en una red, y la salud e integridad de servidores. Cuenta con un sistema flexible de notificaciones que permite a los usuarios configurar alertas basadas en correo electrónico para prácticamente cualquier evento que se produzca, lo que permite una reacción rápida ante problemas en el servidor. Además tiene excelentes capacidades para mostrar informes y visualizar información basada en los datos recopilados, haciéndolo idóneo para planificación de capacidades.

Todos los informes y estadísticas, así como los parámetros de configuración, son accesibles a través de su interfaz basado en web. La interfaz asegura que el estado de la red y los servidores pueda ser accedido desde cualquier ubicación. Correctamente configurado, puede desempeñar una función importante en la monitorización de una infraestructura informática, aunque también es aplicable a pequeñas empresas que cuenta con pocos servidores.

Zabbix es gratuito y software libre. Escrito y distribuido bajo la licencia GPLv2, lo que significa que su código está abierto al público y es libremente distribuible.

Zabbix ofrece:

- Capacidad para auto-detectar servidores y dispositivos de red que se encuentren en la red local.
- Monitorización distribuida con administración centralizada en una web.
- Software de servidor para sistemas compatibles con Unix: Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, y OS X.
- Agentes nativos de alto rendimiento (software agente) para Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, OS X, Tru64/OSF1, Windows NT4.0, Windows 2000, Windows 2003, Windows XP y Windows Vista.
- Monitorización sin agentes.

- Autenticación segura de usuarios.
- Permisos de usuario flexibles.
- Interfaz basada en web.
- Notificaciones flexibles vía correo electrónico de eventos predefinidos.
- Visión a alto nivel de los recursos monitorizados.
- Registro de logs.

Las ventajas que ofrece usar Zabbix frente a otras alternativas en software de monitorización, aparte de su bajo coste de adquisición por ser software libre, es la sencillez de manejo, configuración y puesta a punto con una curva de aprendizaje muy asequible, soporte para SNMP (versiones 1 y 2) y una base de datos centralizada y relacional (Zabbix SIA, 2011).

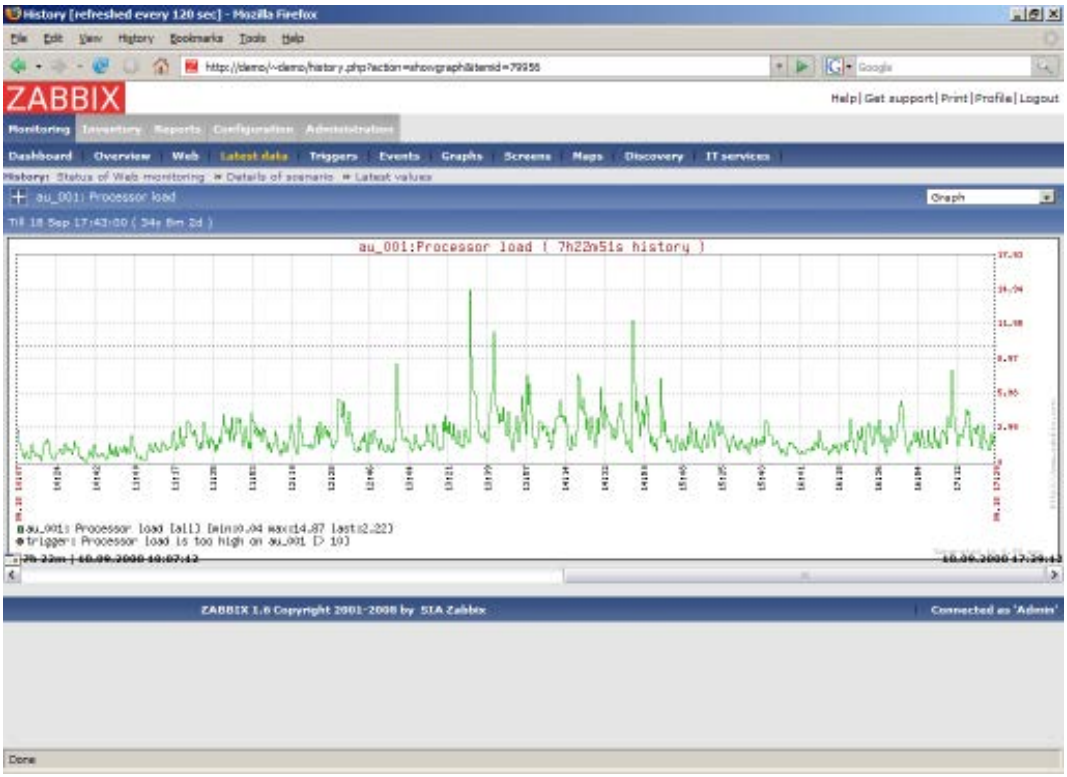

Ilustración 29 Gráfico de carga de CPU en Zabbix

Capítulo III

Herramientas para la elaboración del proyecto

En esta sección se van a presentar las herramientas que se han empleado durante el desarrollo del proyecto, explicando en qué consisten y cuál es la función que se les ha dado.

El proyecto implica distintos tipos de equipos (clientes y servidores), en concreto, tendrá varios servidores para el servicio de Teletrabajo, al menos un servidor para el servicio de monitorización, y por último, máquinas cliente que proveerán los servidores de Teletrabajo y serán monitorizadas.

De modo que este capítulo se estructurará de tal forma que se expliquen las herramientas utilizadas en cada grupo de equipos. Explicando en primer lugar cuáles serán las herramientas implicadas en el desarrollo del servidor de monitorización. A continuación se desarrollarán las herramientas que han sido utilizadas en los servidores de monitorización, para finalizar con los clientes monitorizados.

3.1. Servidor de monitorización

El servidor de monitorización es una única máquina dedicada a la recopilación de datos de otras máquinas que, además, permite presentar al usuario esos mismos datos recopilados. Además, es capaz de notificarle cuando se produzcan eventos definidos previamente, lo que incluye el uso de herramientas para la recopilación de datos y de notificación y comunicación con el usuario. Puesto que en la Oficina de Software Libre contamos con un servidor dedicado a la virtualización de máquinas, se considera interesante desplegar el servidor de monitorización como máquina virtual, ya que esto

ofrece ventajas como la facilidad de despliegue del servidor, flexibilidad en cuanto a su configuración hardware y modificaciones y una mayor seguridad frente a un servicio típico ya que la máquina virtual es independiente de la máquina física.

En este punto se indicará cual es la plataforma de virtualización en la que se montará la máquina virtual que contenga el servidor de monitorización. Para, a continuación, exponer cuál es el sistema operativo escogido para instalarse en la máquina virtual y, por último, detallar la infraestructura software que se desplegará sobre el sistema operativo.

3.1.1. Plataforma de virtualización: Proxmox

Ilustración 30 Logotipo de Proxmox

El objetivo de Proxmox 1.7, según su web (Proxmox Server Solutions GmbH, 2011a), es establecer una infraestructura completa de virtualización en alrededor de una hora. Partiendo desde el hardware, es posible crear una infraestructura empresarial con todas las capacidades incluyendo: proxy de correo, proxy web, wiki, CMS web, intranet, etc., así como backups y migraciones en caliente. Proxmox VE es software libre licenciado bajo la licencia GPLv2.

Proxmox soporta varios tipos de virtualización, entre ellos:

- Virtualización contenida con OpenVZ: es la tecnología preferida para ejecutar servidores Linux al ser la más rápida. OpenVZ crea varios contenedores seguros e independientes, cada uno de los cuales se ejecuta como si fuera un servidor individual al poder reiniciarse independientemente y tiene acceso de root, usuarios, direcciones IP, memoria, procesos, ficheros, aplicaciones, bibliotecas del sistema y ficheros de configuración.
- Virtualización completa con KVM (Kernel-based Virtual Machine): A diferencia de la anterior, aquí cada máquina virtual tiene su propio hardware virtual privado: una tarjeta de red, discos, adaptador gráfico, etc. KVM es perfecto para utilizar sistemas operativos sin modificar, como pueden ser sistemas Windows.

No es necesario disponer de ningún servidor configurado para la instalación de Proxmox, ya que se distribuye con su propio sistema operativo para poder ser instalado directamente en un servidor vacío y en unos minutos tener la plataforma de virtualización en funcionamiento. El disco de instalación de Proxmox instala los siguientes componentes software:

- Sistema operativo completo (Debian Lenny 64).
- Particionamiento del disco duro con LVM2.
- Núcleo del sistema Proxmox VE Kernel que incluye soporte para OpenVZ y KVM.
- Herramientas para la restauración y creación de copias de seguridad.
- Interfaz de administración basada en web.

Proxmox ofrece una interfaz web que permite administrar la plataforma sin necesidad de tener que instalar ninguna herramienta adicional. La interfaz cuenta con una visión de consola de las máquinas virtuales, cuanta con tecnología AJAX para una actualización dinámica de recursos, y permite una conexión segura mediante SSL a las máquinas.

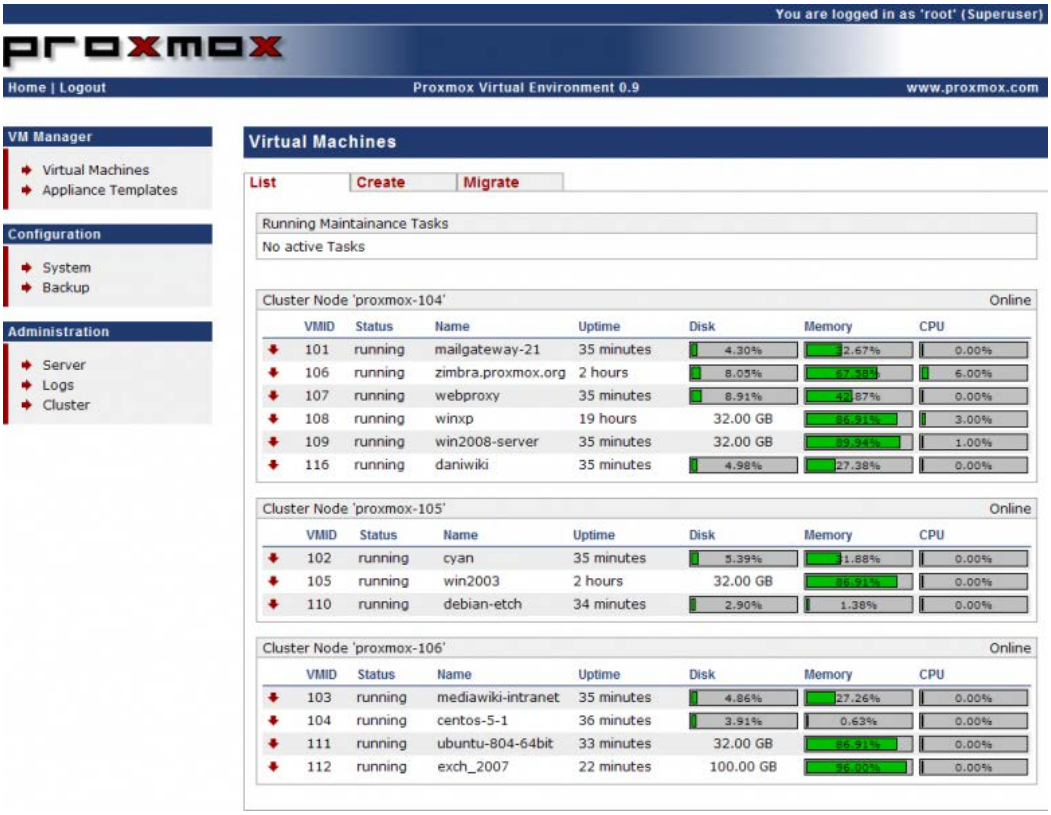

Ilustración 31 Interfáz web de Proxmox

Para la gestión de copias de seguridad Proxmox utiliza VZDump, que es una herramienta que permite realizar copias instantáneas de máquinas que están

funcionando mientras se realiza la copia de seguridad, empaquetándola en un archivo *tar*. Las tareas de copia de seguridad pueden ser gestionadas utilizando la interfaz web de Proxmox.

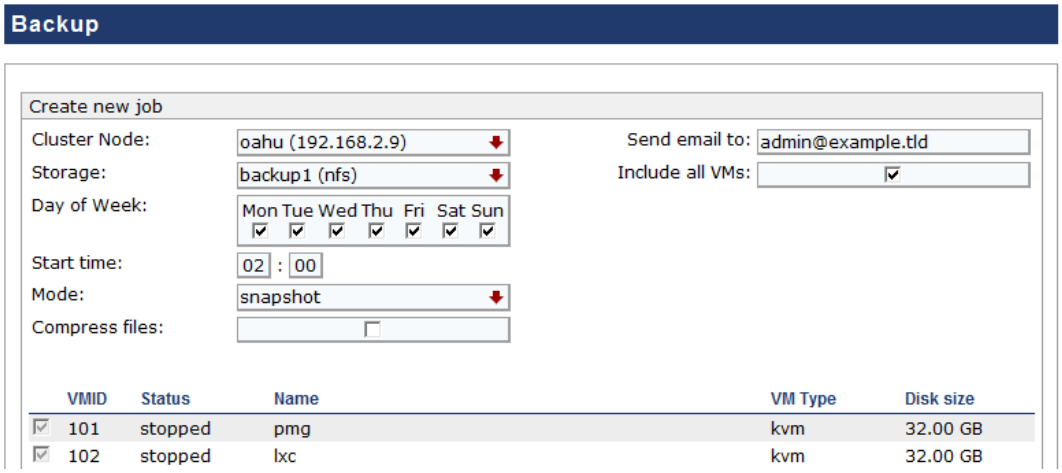

Ilustración 32 Interfáz de administración de backup en Proxmox

Por último, Proxmox ofrece una gestión centralizada de múltiples servidores físicos en clúster, un clúster de Proxmox está constituido por un único servidor maestro y varios nodos. La ventaja que aporta crear un clúster, aparte de un control centralizado y uso del mismo usuario para acceder a los nodos, es que se puede realizar una migración de máquinas entre cualesquiera nodos del clúster, sin necesidad de tener que apagar la máquina migrada (Proxmox Server Solutions GmbH, 2011b).

3.1.2. Sistema operativo: Debian

Los ordenadores modernos constan de una gran cantidad de dispositivos o periféricos (discos, pantallas, interfaces de red…) que los convierten en un sistema de gran complejidad. El programador que escriba un programa para ese sistema se encuentra con una tarea extremadamente difícil ya que debe tener en cuenta cada uno de esos componentes y utilizarlos aunque no sea de forma óptima. Por lo tanto se utilizan sistemas operativos, que son una capa de software encargada de administrar esos dispositivos ofreciendo a los programas una interfaz simplificada para que se puedan comunicar de una forma más sencilla con los dispositivos (Tanenbaum, 2003).

Ilustración 33 Logotipo de Debian

El proyecto Debian fue fundado el 16 de agosto de 1993 por Ian Murdock, un estudiante de la Universidad de Purdue y fue patrocinado por la GNU hasta noviembre de 1995, desde el principio se ideó como un proyecto colaborativo propio del proyecto GNU en lugar de ser un proyecto de una única persona, hasta la actualidad que cuenta con el millar de desarrolladores voluntarios.

Debian es una distribución de GNU/Linux (aunque se están desarrollando versiones sin Linux que utilizan el núcleo Hurd de GNU o el núcleo de BSD) que actualmente incluye más de 29000 paquetes de software y herramientas para que el usuario pueda personalizarlo escogiendo cuales de estos paquetes instalar (Proyecto Debian, 2011). Al basarse en licencias de software libre (principalmente la GPL de GNU) es totalmente libre de distribuir y gratuito, Debian es una de las distribuciones de GNU/Linux que únicamente distribuyen software libre, no significando que no se pueda usar software de código no libre, para lo cual permite incluir entre sus fuentes de software el repositorio *non-free* y *contrib* con paquetes cuya distribución o dependencias están limitadas.

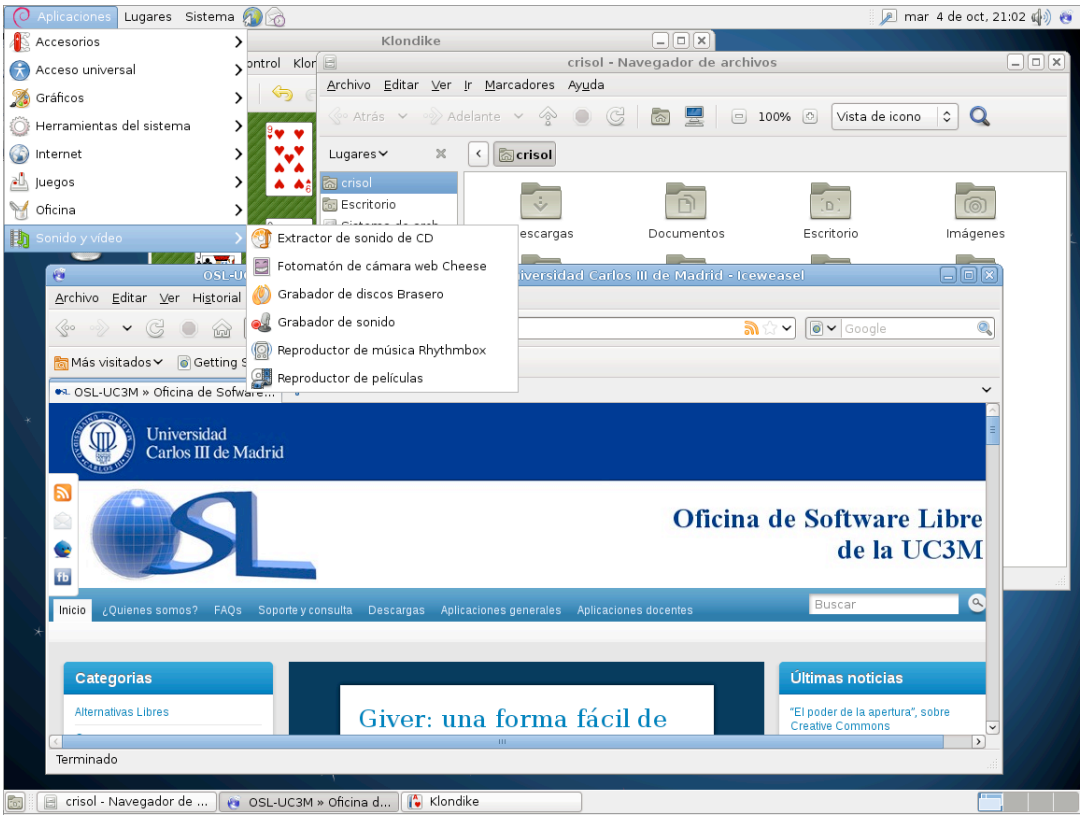

Ilustración 34 Sesión de escritorio en Debian

En la ilustración se puede observar una máquina virtual i686 funcionando bajo el sistema operativo GNU/Linux Debian 6.0.2 (Squeeze) con el entorno de escrito *Gnome* 2, que trae por defecto la distribución, y su característico navegador web *IceWeasel* derivado de *Firefox* (dado que *Firefox* no puede ser incluido en Debian debido a la estricta política del proyecto de únicamente incluir software totalmente libre).

3.1.3. Infraestructura software

A continuación se listará el software más relevante (puesto que se usa mucho más, como el interfaz gráfico del sistema, gestor de paquetes, editor de texto, etc.) que estarán instaladas en la máquina que hace de servidor central de monitorización. El software presentado a continuación puede, como es el caso de Zabbix o del interfaz gráfico del sistema, utilizarse también en las máquinas clientes.

3.1.3.1. Apache HTTP Server

Ilustración 35 Logotipo de la Apache Software Fundation

El proyecto Apache HTTP Server es un desarrollo colaborativo de software que pretende crear un servidor web robusto capaz de competir con cualquier servidor comercial pero de código libre. El proyecto está contenido en la Apache Software Foundation y lo llevan a cabo un grupo de voluntarios de todas partes del mundo.

HTTP daemon (httpd), creado por Rob McCool en la NCSA (National Center for Supercomputing Applications), era el servidor web más popular en 1995, pero su desarrollo se estancó a mediados de 1994 cuando Rob McCool dejó la NCSA. Fue entonces cuando un grupo de usuarios de httpd se pusieron en contacto entre sí para coordinarse en el desarrollo de parches y mejoras para el servidor, así se crea el Apache Group formado por Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau, Andrew Wilson. A partir de las mejoras y corrección de errores que hicieron de la versión 1.3 de httpd se hizo el primer lanzamiento de la versión 0.6.2 de Apache server en 1995 (The Apache Software Foundation, 2011).

Algunas de las características de Apache HTTP Server son las siguientes:

- Es software libre.
- Implementa los últimos protocolos, como HTTP/1.1.
- Se puede configurar y ampliar mediante módulos desarrollados por terceras partes.
- Ofrece un API y todo su código fuente para poder crear módulos y modificaciones.
- Hay versiones para la mayoría de sistemas Unix, además de Windows 2000, OS/2 o Netware 5.X.
- Es un proyecto activo, en el que la comunidad sigue implementando o sugiriendo mejoras y corrigiendo errores.
- Acceso mediante contraseña a determinadas páginas web.
- Permite servidores virtuales, es decir, una misma máquina puede distinguir y procesar peticiones hechas a direcciones distintas.

- Apache HTTP Server es un servidor ampliamente utilizado habiendo en abril de 2010 más de 120 millones de servidores, haciendo que haya sido exhaustivamente probado, no solo por los usuarios, sino por el propio Apache HTTP Server Project Management Committee.

3.1.3.2. PHP

Ilustración 36 Logotipo de PHP

PHP acrónimo de "PHP: Hypertext Procesor" es un lenguaje de programación interpretado de alto nivel de código libre. Se utiliza principalmente en páginas web dotándolas de contenido dinámico, para lo cual, se puede incrustar el código escrito en PHP dentro de un HTML.

El origen de PHP se remonta a 1995 cuando Rasmus Lerdorf crea PHP/FI (Personal Home Page Tools), un conjunto de scripts de Perl que utilizó para controlar el acceso a su página web. Con el tiempo fue escribiendo código en C para incrementar su funcionalidad, y tras arreglar el código de fallos y prepararlo, decide liberar el código fuente.

A diferencia del otro lenguaje más extendido de programación utilizado en web: JavaScript, PHP no se ejecuta en el equipo del cliente que visualiza la página, en cambio lo hace en el servidor que la sirve, de modo que el cliente no percibe ese código, obteniendo únicamente el resultado de su ejecución en forma de un HTML como cualquier otro en el navegador.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales (Achour *et al.*, 2011).

3.1.3.3. Exim

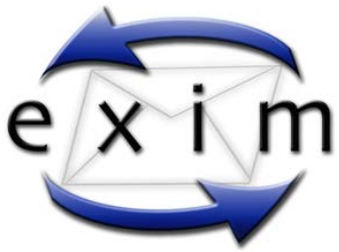

Ilustración 37 Logotipo de Exim

Exim es una MTA (mail transfer agent) alternativa a Sendmail en sistemas compatibles con Unix. Es software libre licenciado bajo la licencia GNU GPL. Fue creado en la Universidad de Cambridge por Philip Hazel en 1995 para sustituir los entonces utilizados servidores de correo del University of Cambridge Computing Service Smail y Sendmail durante la transición de las redes internas de la universidad X.25 a la nueva red pública que es Internet. En principio Exim no sería más que una ampliación del ya mencionado Smail, pero por aquél entonces el código de Smail ya era antiguo, comenzando a escribir código nuevo incrementando la funcionalidad de Smail y eliminando el soporte de UUCP que en principio creía innecesario para sus necesidades y lo bautizó como EXperimental Internet Mailer. Originalmente se desarrolló para funcionar en servidores de tamaño mediano, con una conexión permanente de red y en un ámbito universitario, no obstante la difusión que ha tenido desde que un ISP de Reino Unido se ofreció para dedicarle una página y web y una lista de correo, lo ha hecho adaptarse a una variedad de máquinas, desde ordenadores personales hasta clusters con millones de usuarios.

Las MTA se encargan de recibir mensajes desde distintas fuentes y transmitirlos por el lugar adecuado para que lleguen a su destinatario. En concreto Exim recibe los mensajes remotamente mediante el protocolo SMTP y localmente directamente de los procesos, pudiendo enviar directamente los mensajes al mailbox del sistema.

A continuación se detallan algunas de las características de Exim:

- Filosofía: Exim está diseñado para usarse en redes donde la mayoría de mensajes son entregados en el primer intento (como es el caso de la red universitaria en el que fue creado), por lo que no es necesario un complejo mecanismo centralizado de colas por el que pasan los mensajes. De modo que el funcionamiento por defecto de Exim es

enviar el mensaje nada más recibirlo, lo que conlleva problemas de retardo en la entrega cuando recibe varias conexiones SMTP simultáneas, ya que el servidor desactiva la entrega inmediata y encola los mensajes para enviarlos en otro momento.

- Colas: A diferencia de otros servidores de correo, Exim en lugar de mantener una cola distinta para diferentes equipos o dominios, solo tiene una en la que almacena todos los mensajes, y a diferencia de una cola normal, en el que el primer mensaje en entrar es el primero en enviarse, es estrictamente una colección en el que el orden de los elementos en la estructura no se corresponde con el de llegada de los mensajes.
- Procesos: Se puede lograr paralelismos mediante el uso de distintos procesos, en Exim no existe ningún proceso central que sirva para coordinar las acciones de Exim, no habiendo manera de iniciar o detener el servicio Exim. Son el resto de procesos los que inician un nuevo proceso Exim cuando lo necesitan.

(Hazel, 2003)

3.1.3.4. MySQL

Ilustración 38 Logotipo de MySQL

MySQL es un sistema de administración de bases de datos relacionales rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, en especial en arquitecturas cliente/servidor. Es el más célebre sistema gestor de bases de datos del mundo Open Source, gracias a su compatibilidad con el servidor HTTP Apache y PHP (Thibaud, 2006).

En 1979 Monty Widenius, trabajando para una pequeña compañía llamada TcX, creo una herramienta de informes escrita en BASIC para ejecutarla en un ordenador de 4Mhz con 16KB de RAM. Con el tiempo la herramienta se reescribió en C y fue portada a Unix, no era más que un motor de almacenamiento a bajo nivel con una interfaz para obtener informes y fue bautizada como Unireg.

A principios de la década de 1990, los clientes de TcX pidieron disponer de una interfaz SQL para acceder a su información, Monty planteo el uso de una solución comercial de base de datos, pero no estaba satisfecho con la velocidad de esa solución, de modo que intentó adaptar el código de mSQL para utilizarse con Unireg, lo que tampoco llegó a funcionar bien. Fue entonces cuando decidió escribir el código que necesitaba por sí mismo. Lanzando de esta manera la versión 1.0 de MySQL en mayo de 1996 a un grupo limitado y posteriormente en octubre de 1996 se lanzó públicamente la versión 3.1.11 (Alexander Pachev y Sasha Pachev, 2007).

Las características de MySQL son (Reese *et al.*, 2002):

- Abierto: MySQL es de código abierto desde su fundación, así como lo es el dialecto de SQL que utiliza denominado ANSI SQL2. Su motor de base de datos esta portado a una gran cantidad de plataformas como Windows 2000, Mac OS X, Linux, FreeBSD y Solaris. Aunque si no está disponible para una plataforma concreta puede ser compilado desde su código fuente al estar esté disponible.
- Soporte de aplicaciones: MySQL ofrece una API para prácticamente cualquier lenguaje de programación. En concreto se pueden escribir programas que accedan a la base de datos MySQL en C, C++. Eiffel, Java, Pearl, PHP, Python y Tcl.
- Unión de bases de datos cruzada: las consultas en MySQL se pueden hacer consultando tablas de distintas bases de datos.
- Soporte para unión de bases de datos externas: MySQL puede hacer consultas implicando bases de datos externas utilizando sintaxis ANSI y ODBC.
- Internacionalización: soporta distintos juegos de caracteres, incluyendo ISO-8859-1, Big5 y Shift-JIS. Permite realizar ordenaciones mediante juegos de caracteres diferentes. Los mensajes de error se ofrecen en varios lenguajes.

- Relación calidad/precio: el gran beneficio que aporta utilizar MySQL es que es muy rápida, obteniendo un gran rendimiento, a un coste bajo.

3.1.3.5. MySQLTuner

MySQLTuner es un script escrito en Perl para revisar rápidamente una instalación de MySQL y hacer ajustes que incrementen su rendimiento o estabilidad. El script presenta en pantalla el estado las variables tal como están configuradas en la base de dato, junto con sugerencias para optimizar el rendimiento.

Los requisitos para poder utilizar MySQLTuner es disponer de una máquina con sistema operativo compatible con Unix (BSD, Linux o Solaris) con Perl 5.6 o posterior para lanzarlo, aunque puede analizar remotamente la base de datos. Las versiones de MySQL totalmente soportadas van desde la 3.23 a la 5.1, las versiones posteriores solo tienen un soporte parcial.

3.1.3.6. Servidor Zabbix

De entre todas las alternativas de software de monitorización con software libre, se ha escogido implantar un sistema Zabbix (*2.5.4 Zabbix*). Los motivos por los que resulta más interesante Zabbix frente a las otras alternativas son principalmente:

1. Ofrece una gran cantidad de características, como son las plantillas, configuración remota de agentes o comprobaciones transaccionales de sitios web, que Pandora FMS *2.5.3 Pandora FMS (Flexible Monitoring System)* únicamente ofrece en su versión privativa de pago.
2. Instalación sencilla al tratarse de un único componente que presenta toda la funcionalidad y sin necesidad de añadir módulos para cada nuevo elemento que se desee monitorizar, en oposición a Nagios (*2.5.2 Nagios*), el cual, debido a su modularidad, requiere un esfuerzo añadido para descubrir que módulos son los que se necesitan utilizar en nuestro caso e incluirlos en el sistema y los agentes.
3. Zabbix muestra los valores monitorizados en formato numérico, pudiendo conocer en cada momento el valor exacto que presenta un

elemento monitorizado. Esta característica es importante a la hora de valorar el rendimiento de las máquinas virtuales monitorizadas para poder ajustarlas a las necesidades de los usuarios. Nagios en cambio únicamente ofrece valores nominales, que pese a simplificar las tareas de mantenimiento del administrador de las máquinas monitorizadas, resultan inútiles para nuestro propósito.

Zabbix es un sistema distribuido que consta de dos partes principalmente: el proceso servidor, que es el que se localizará en el servidor de monitorización, y los agentes, que son los procesos localizados en los clientes monitorizados.

3.2. Servidor de teletrabajo

Los servidores de teletrabajo, puesto que son el lugar donde se localizan los clientes monitorizados y el punto que centraliza las conexiones de red de ellos, se hace el lugar indicado para establecer los servicios de seguridad necesarios.

Primero se hablará sobre el software que se ha utilizado para realizar el control de acceso, tanto hacia los clientes monitorizados como para los servidores que los hospedan, y se finalizará este punto con el entorno de programación en el que se ha desarrollado la funcionalidad de disponibilidad de datos y recursos.

3.2.1.1. Shorewall

Ilustración 39 Logotipo de Shorewall

Shorelin Firewall, comúnmente conocido como “Shorewall”, es una herramienta de alto nivel para configurar Netfilter, permite definir las entradas del firewall mediante un conjunto de ficheros de configuración. Shorewall lee todos esos ficheros de configuración y mediante las utilidades iptables, iptables-restore, ip y tc, configura Netfilter y el subsistema de redes de Linux. Shorewall puede ser utilizado en un sistema de firewall dedicado, en una puerta de enlace o enrutador, servidor o simplemente en un sistema GNU/Linux.

A diferencia de un firewall, Shorewall no es un proceso del sistema, cuando se lanza el programa, configura el subsistema de redes de Linux y termina, por lo que el proceso Shorewall no queda en ejecución en el sistema.

No es una herramienta sencilla de utilizar en cuanto a las existentes para la configuración de Iptables, pero a cambio, es la más potente y flexible. La dificultad que entraña, aparte de no disponer de una interfaz gráfica para la configuración del firewall, es que requiere un mínimo de conocimiento sobre redes, por lo que no es una herramienta indicada para cualquier usuario, pero es idónea para manejar entornos de red complejos que cambian rápidamente.

La configuración de Shorewall se hace mediante los ficheros de texto plano localizados en "/etc/shorewall/", "zones", "interfaces", "policy", "rules" y "hosts". Una vez definidos estos ficheros, hay que compilar Shorewall, proceso mediante el cual se lee la configuración de los ficheros y genera un script de Bash que ejecuta automáticamente. Si hay algún fallo en la definición de los ficheros, el proceso de compilación se aborta descartando el script.

Shorewall es software libre, estando permitida su redistribución o modificación según la licencia GPLv2 publicada en la Free Software Foundation (Eastep, 2009).

3.2.1.2. Bash

Bash es la shell (o intérprete de comandos) del sistema operativo GNU, Bash es un acrónimo de Bourne-Again SHell, haciendo un juego de palabras con el nombre de la shell de Unix 7 Bourne Shell (creada en 1977 por Stephen Bourne) indicando su renacimiento. Comenzó su desarrollo por Brian Fox del Free Software Foundation en enero de 1988 y publicada la primera beta en junio de 1989.

Bash es compatible con sh e incorpora características de las shells Korn (ksh) y C (csh). Es una implementación los estándares IEEE POSIX P1003.2/ISO 9945.2 sobre shells y herramientas, aportando mejoras funcionales sobre el original sh tanto para programación como uso, además, casi todos los scripts de sh pueden ejecutarse en Bash sin necesidad de ser modificados (The GNU Project, 2009)

Entre las características de Bash según (Ramey, 2011) destacan las siguientes:

- Edición de líneas de comandos de una forma similar a como se haría en emacs o Vi. De manera que no es necesario eliminar todo el comando hasta el punto que se quiere corregir, o eliminar el comando

entero para volver a escribirlo. Además tiene una función de auto-completado para comandos del sistema o nombres y rutas de ficheros.

- Histórico de comandos ejecutados ilimitado, recordando los comandos introducidos para permitir al usuario recuperarlos y volver a usarlos o editarlos para una nueva situación. Es configurable que comandos se desee que se recuerde.
- Control de tareas: Bash ofrece una interfaz al control de tareas del sistema operativo, que permite suspender, reiniciar o mover entre primer y segundo plano una tarea, permitiendo a los usuarios que se "olviden" de las aplicaciones en segundo plano.
- Alias, consistentes en definir cadenas de texto que sustituyen un comando facilitando u ocultando la ejecución de un comando más complejo.
- A partir de la versión 2.0 de Bash se pueden indexar arrays de tamaño ilimitado, ofreciendo distintos tipos de arrays ya predefinidos.
- Permite realizar operaciones aritméticas con números enteros en distintas bases, desde base 2 a base 64, ofreciendo casi todos los operadores que ofrece C y con la misma sintaxis. Se pueden utilizar variables como operandos y se puede guardar el resultado en variables.
- Control del comportamiento de la shell mediante propiedades y variables, y personalización del propt.
- Capacidades ampliadas de entrada/salida, Bash permite especificar un descriptor de fichero o tunería como entrada o salida.
- Bash ofrece una pila de directorios en la que se añaden y eliminan directorios durante la sesión. Haciendo que sea muy sencillo cambiar el directorio actual por otro que se encuentre en la pila.
- Permite adaptarse al idioma del sistema traduciendo las salidas y utiliza caracteres de 8 bits para las entras, permitiendo la mayoría de las familias de caracteres ISO-8859.

3.3. Clientes monitorizados

Por último, se detallará el software implicado en esta parte del proyecto para los clientes monitorizados. Desde la perspectiva que se tiene en estos momentos, estos

equipos no son más que sistemas que hay que monitorizar, sobre los que tenemos un control y acceso limitado al software que ejecutan. Debido a esto, únicamente se explicará el software que presentan estas máquinas para poder ser monitorizadas, así como la herramienta que se ha utilizado para realizar las pruebas de rendimiento, para asegurar el funcionamiento de los equipos ante situaciones de alta carga del sistema.

Una visión más amplia de las herramientas utilizadas para estos equipos puede obtenerse en el proyecto predecesor a este (Gil Bázquez, 2011).

3.3.1. Agente de Zabbix

Como se ha explicado en el punto *3.1.3.6 Servidor Zabbix*. Zabbix (*2.5.4 Zabbix*) es un sistema distribuido que cuenta principalmente con un proceso servidor y varios procesos cliente. En cada uno de estos equipos monitorizados se instalará uno de los clientes de Zabbix denominados agentes de Zabbix.

Los agentes de Zabbix son un proceso del sistema encargado de obtener datos del equipo en él se ejecutan. Para la recopilación de estos datos pueden utilizarse varios métodos, como utilizar llamadas del sistema, ejecutar programas, o leer ficheros de log. La recopilación de datos se realiza de una forma iterativa con periodos configurables por el administrador y de duración variable para cada uno de los elementos que se desea monitorizar. Cada vez que se recopila un dato, este se envía al servidor de Zabbix indicado en la configuración del agente.

Los agentes de Zabbix permiten realizar la monitorización de dos formas distintas: una primera en el que el agente es un proceso pasivo que se limita a recibir peticiones de que elemento monitorizar, a las que responde con el valor medido, y una segunda forma en la que el agente es un proceso activo que por iniciativa propia recopila los datos necesarios y los envía al servidor.

3.3.2. HeavyLoad

Ilustración 40 Logotipo de HeavyLoad

HeavyLoad es la única excepción en cuanto software libre de este proyecto, ya que se distribuye como freeware por la empresa alemana Jam Software.

La función de HeavyLoad es provocar estrés de diferentes recursos de un ordenador (como CPU, memoria RAM, disco duro, sistema operativo, etc.) con el objetivo de probar la fiabilidad de dicha máquinas ante situaciones de una alta carga. Lo que es útil para evaluar distintas alternativas de ficheros o servidores de bases de datos antes de utilizarlos en producción, o simplemente para comprobar si la temperatura del ordenador se eleva demasiado en casos de un uso intensivo.

HeavyLoad también sirve para probar el comportamiento de la máquina en sistemas con recursos limitados (memoria, espacio en disco) (JAM Software GmbH, 2011).

Las opciones que ofrece HeavyLoad para probar el estrés de la máquina son:

Carga de CPU: Este test carga completamente la capacidad del procesador, mediante el uso de cálculos complejos.

Escritura en fichero temporal: HeavyLoad crea un fichero temporal en el que escribe continuamente información durante la prueba. La tasa de información a la que escribe en el fichero puede ser definida en el diálogo de opciones.

Ocupación de memoria: HeavyLoad ocupa memoria continuamente de manera que el sistema tenga cada vez menos memoria disponible. La tasa con la que la memoria será ocupada con datos puede ser definida en las opciones.

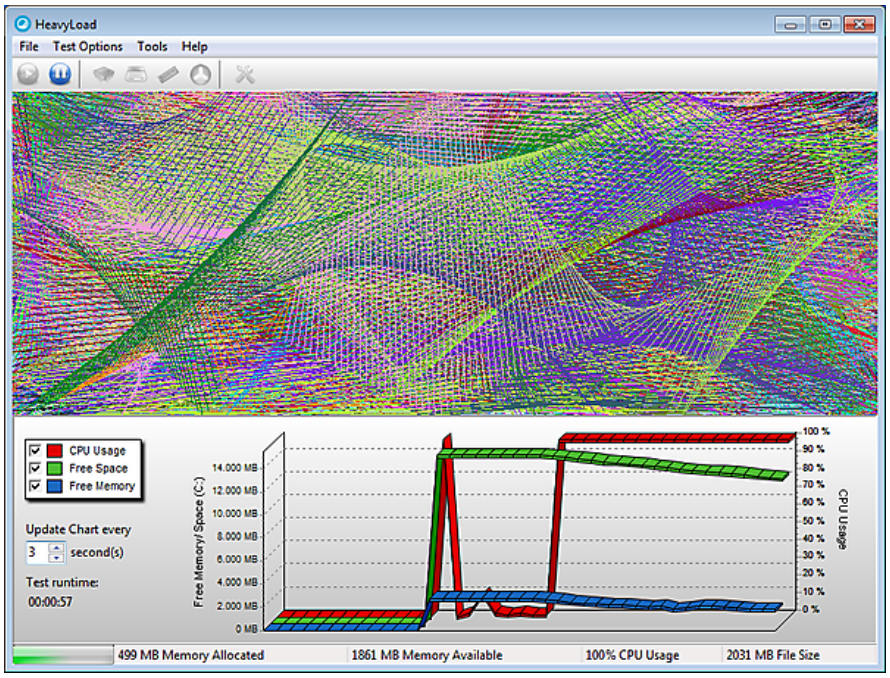

Ilustración 41 HeavyLoad realizando todas las pruebas a la vez en una máquina

Capítulo IV

Desarrollo del proyecto

A lo largo de este capítulo se desarrollará el proceso que se ha seguido desde el momento que surge la necesidad de ofrecer un servicio de seguridad que proteja al Servicio de Teletrabajo hasta la construcción y mantenimiento del mismo.

En primer lugar se intentará decidir cuál es el ciclo de vida más apropiado para el desarrollo del proyecto según las características del mismo y de las restricciones que sean impuestas. El ciclo de vida software escogido definirá unas fases por las que debe pasar el proceso de producción que se corresponderán con los puntos que tratará este capítulo.

4.1. Fase inicial

Antes de comenzar el desarrollo del proyecto hay que decidir el ciclo de vida del software que se seguirá, definiéndonos cuáles serán las fases por las que debe pasar el proyecto para llegar a finalizarlo.

Al igual que en la primera mitad del proyecto desarrollado (Gil Bázquez, 2011), y por coherencia con el mismo, se continúa con el mismo **ciclo de vida en cascada** utilizado anteriormente. El desarrollo de este proyecto se ha realizado, en gran medida, en paralelo con el proyecto anterior, conteniendo muchas partes del desarrollo en común. Por este motivo se hace razonable seguir el mismo ciclo de vida que ha seguido ese proyecto, en lugar de utilizar otro distinto que dificulte la interacción entre ambos. El desarrollo de partes como el backup o la monitorización, requieren de experimentación y rediseño, para poder llegar a una solución óptima en cuanto al backup o para decidir qué es lo que realmente se quiere monitorizar. De este modo, también eran apropiados ciclos de vida basados en prototipos o iterativos, que permitieran la vuelta atrás para realizar un

rediseño. No obstante, el ciclo de vida en cascada permite esta vuelta atrás ante errores, siendo posible su uso sin ningún problema importante.

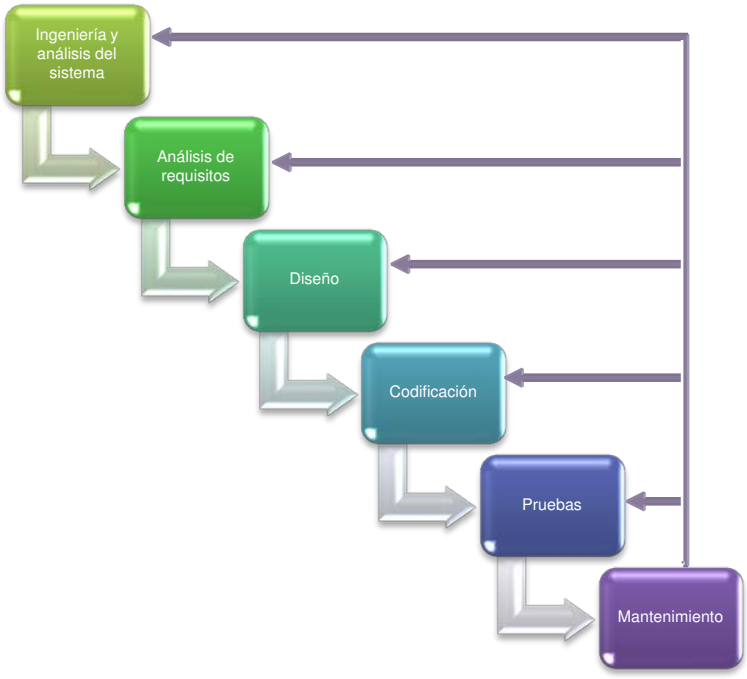

Ilustración 42 Ciclo de vida en cascada

En la *Ilustración 42 Ciclo de vida en cascada* se observa como este ciclo de vida define seis fases diferentes, aunque en este proyecto se realizarán en una única fase las de "ingeniería y análisis del sistema" y "Análisis de requisitos" por simplicidad llamada simplemente "Análisis".

En primer lugar se comenzará con la fase de análisis que recoge y representa la funcionalidad del sistema mediante diagramas de casos de uso, para la posterior extracción de requisitos. Para realizar el análisis y definir cuál es la funcionalidad que se desea ofrecer, nos basaremos en la funcionalidad que ofrecen los actuales sistemas de monitorización en el mercado, también se reflejarán las restricciones que nos son impuestas como, por ejemplo, las copias de seguridad del Servicio de Teletrabajo.

Seguidamente, en la fase de diseño se detallará el control de acceso, definiendo cuáles son las políticas que requiere el sistema. En este punto, cuando tengamos un diseño de los equipos implicadas en el sistema, será preciso definir qué conexiones se permiten y cuáles no. También en este punto se decidirá qué elementos monitorizar del sistema y, finalmente, cómo y cuándo se realizará el backup.

La fase de codificación implicará, en cuanto a implementación, el desarrollo de los scripts necesarios para articular el diseño realizado en el apartado anterior. Además, en esta fase es donde se realizará el despliegue de equipos e implantación del software.

Durante la fase de pruebas se comprobará que el sistema satisface todos los requisitos especificados durante la fase de análisis, y se realizarán pruebas que midan el rendimiento del sistema, subsanando las incidencias detectadas en esta fase, relacionadas con el incumplimiento de requisitos no funcionales del sistema.

Para la última fase, mantenimiento, se elaborarán unos guiones que describan los procedimientos a seguir para asegurar el correcto funcionamiento del Servicio de Teletrabajo dentro de los niveles de servicio establecidos.

Como se ha mencionado anteriormente, parte del desarrollo de la funcionalidad del proyecto se ha realizado de forma iterativa, concretamente, la definición de los elementos a monitorizar y el ajuste de los umbrales que harán saltar las alertas. En estos casos, se iterará entre la fase de diseño y codificación. Por simplicidad, en este documento se describirá el proceso resumido como una única iteración por cada fase, para una mayor claridad.

4.2. Análisis del sistema

La fase de análisis del ciclo de vida pretende especificar qué es lo que queremos que realicen los sistemas que se van a desarrollar en este proyecto, dándonos un punto de partida con las características que se deben implementar en un futuro, definidas por los requisitos de software. Además, estos requisitos nos sirven como un guión a lo largo del proyecto ofreciendo unas pautas del orden en el que se deberá ir desarrollando la funcionalidad y posteriormente, al final del proyecto, se utilizarán para verificar que realmente el sistema ha cumplido con los objetivos establecidos en cada uno de los requisitos.

En primer lugar se definirán diagramas de casos de uso que sirvan de apoyo para poder determinar cuál es la funcionalidad deseada y con esa base poder comenzar con la extracción de requisitos de usuario.

4.2.1. Diagrama de casos de uso

En este apartado se pasará a definir una serie de casos de uso modelados según UML, que indicarán que es lo que deseamos que hagan los sistemas involucrados en el proyecto, lo que supone un punto de partida para el análisis del proyecto que nos ayude más adelante en el proceso de extracción de requisitos.

Puesto que este proyecto abarca dos sistemas, uno completamente (el servicio monitorización) y otro parcialmente (servicio de teletrabajo). Se mostrarán todos los

casos de uso de ambos sistemas, pero aparecerán en gris los que quedan fuera de este proyecto concreto y están completamente detallados en el proyecto anterior (Gil Bázquez, 2011).

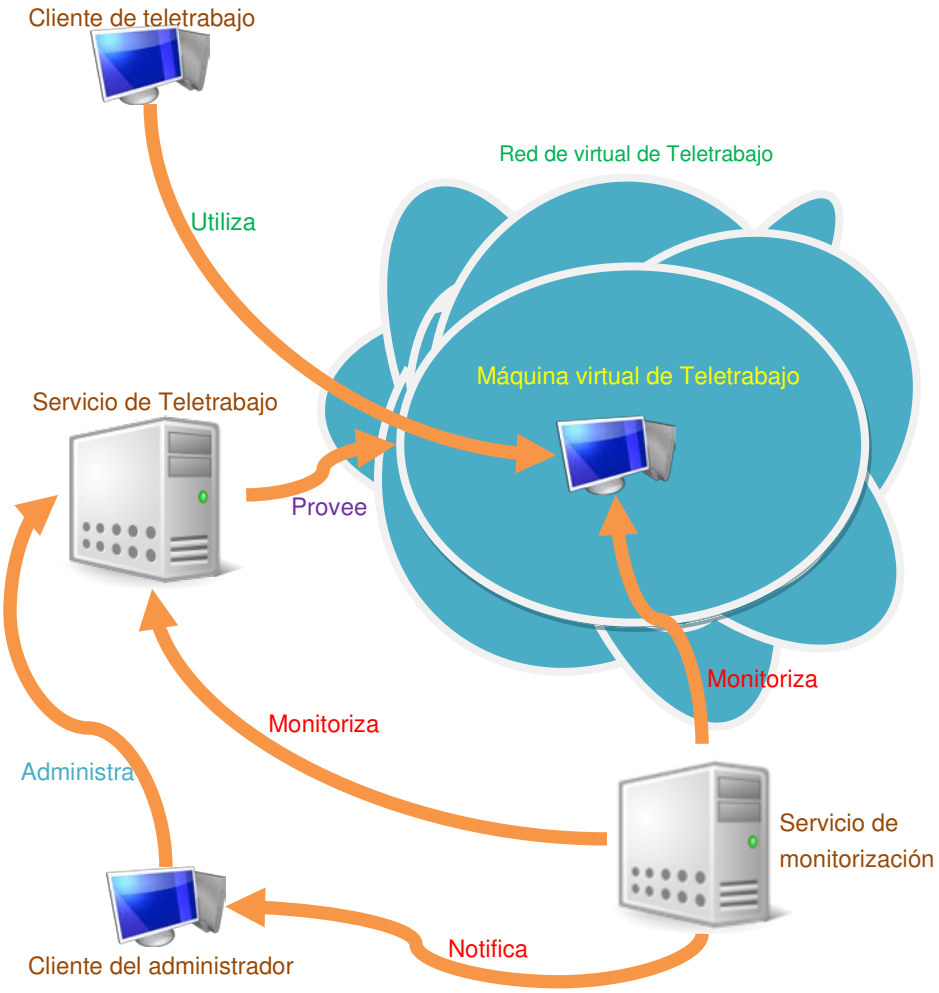

Ilustración 43 Esquema de las relaciones entre los sistemas

En *Ilustración 43 Esquema de las relaciones entre los sistemas* se puede ver como se tienen dos servicios independientes: el de Teletrabajo, encargado de proveer máquinas virtuales para que el cliente de teletrabajo se conecte a ellas mediante escritorio remoto, por otra parte está el servicio de monitorización, que se encargará de supervisar el funcionamiento tanto del servidor de Teletrabajo como de las máquinas virtualizadas, e informar a un administrador si se detecta algún problema.

En primer lugar se especifican los casos de uso que afectan al Servicio de Teletrabajo que se muestran en la figura a continuación.

4.2.1.1. Casos de uso del servicio de Teletrabajo

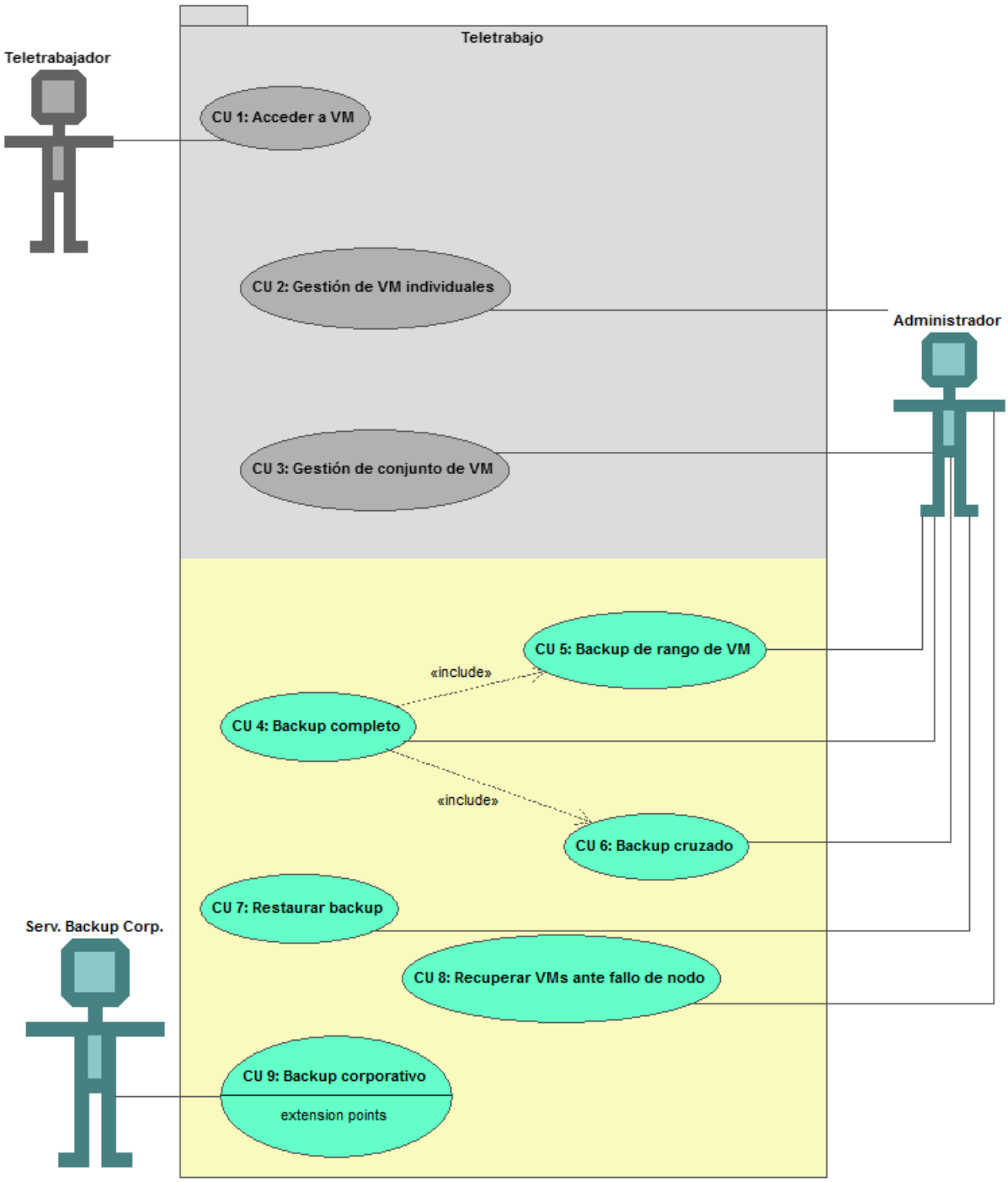

Ilustración 44 Diagrama de casos de uso del Servicio de Teletrabajo

Se puede observar que disponemos de tres actores diferenciados: el **teletrabajador**, que es el usuario final del Servicio de Teletrabajo y que accede a su máquina virtual y trabaja con ella. Luego tenemos al **administrador** del

Servicio de Teletrabajo, que es la persona encargada de mantener el Servicio de Teletrabajo, gestionando las máquinas y recuperándolas ante cualquier problema. Por último está el rol que denominaremos **Serv. Backup Corp.**, el cual representa otros servicios que se ofrecen en la Universidad Carlos III, que en el caso que ocupa, es el *servicio de backup corporativo* de la Universidad.

Identificador: CU 1	
Nombre:	Acceder a VM.
Actores:	Teletrabajador.
Objetivo:	-
Precondiciones:	El teletrabajador tiene una conexión a internet. El teletrabajador está conectado a la VPN de la UC3M.
Postcondiciones:	El teletrabajador puede trabajar con su máquina virtual remotamente.
Escenario básico:	1. El teletrabajador accede a la pantalla de conexión de escritorio remoto. 2. El teletrabajador introduce el nombre o la dirección IP de su máquina virtual. 3. El teletrabajador inicia la conexión con su máquina virtual. 4. El teletrabajador introduce sus datos de acceso a la máquina virtual para iniciar sesión. 5. El teletrabajador accede a su máquina virtual.
Escenarios alternativos:	2a. El teletrabajador introduce un nombre o dirección IP de máquina virtual errónea o se produce un error en la red. 1. No se puede producir la conexión con la máquina virtual. 2. Volver al paso 1. 4a. El teletrabajador introduce sus datos de inicio de

Identificador: CU 1	
	sesión en la máquina virtual de manera errónea. 1. No se puede iniciar sesión en la máquina virtual. 2. Volver al paso 3.

Tabla 1: Definición del caso de uso CU 1

Identificador: CU 2	
Nombre:	Gestión de VM individuales.
Actores:	Administrador.
Objetivo:	Gestionar las máquinas virtuales de manera individual.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha realizado una acción sobre una máquina virtual concreta.
Escenario básico:	1. El administrador accede a la pantalla de administración de máquinas virtuales. 2. El administrador selecciona una máquina virtual concreta. 3. El administrador efectúa una operación sobre esta máquina virtual. 4. La acción requerida por el administrador sobre la máquina virtual ha sido efectuada con éxito.
Escenarios alternativos:	3a. El administrador escoge una máquina virtual errónea. 1. La acción termina con error. 2. Volver al paso 1. 3b. El administrador efectúa una operación inválida sobre la máquina virtual seleccionada debido a su estado actual. 1. La acción termina con error.

Identificador: CU 2	
	2. Volver al paso 1.

Tabla 2: Definición del caso de uso CU 2

Identificador: CU 3	
Nombre:	Gestión de conjunto de VM.
Actores:	Administrador.
Objetivo:	Gestionar un conjunto de máquinas virtuales simultáneamente.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha realizado una acción sobre un conjunto de máquinas virtuales.
Escenario básico:	1. El administrador accede a la pantalla de administración de máquinas virtuales. 2. El administrador selecciona un conjunto de máquinas virtuales. 3. El administrador efectúa una operación sobre este conjunto de máquinas virtuales. 4. Terminar.
Escenarios alternativos:	3a. El administrador escoge un rango de máquinas virtuales erróneo. 1. La acción termina con error para las máquinas virtuales inexistentes pero se completa para el resto de máquinas virtuales. 2. Volver al paso 1. 3b. El administrador efectúa una operación inválida para alguna de las máquinas virtuales seleccionadas debido a su estado actual. 1. La acción termina con error para esa máquina virtual pero se completa para las máquinas que la soporten. 2. Volver al paso 1.

Tabla 3: Definición del caso de uso CU 3

Identificador: CU 4	
Nombre:	Backup completo.
Actores:	Administrador.
Objetivo:	Se realiza una copia de seguridad de todas las máquinas virtuales de un rango establecido, eliminando las copias de seguridad de máquinas antiguas que superen un cierto número de días de antigüedad, así como una copia de los ficheros de configuración de todas las máquinas virtuales de un nodo en el otro y viceversa.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha realizado una copia de seguridad de cada una de las máquinas de un rango concreto, se ha comprobado, y eliminado si procedía, las copias de seguridad de máquinas virtuales antiguas y se ha realizado una copia cruzada de los ficheros de configuración de todas las máquinas virtuales en cada nodo.
Escenario básico:	1. El administrador accede a la pantalla de administración de copias de seguridad de máquinas virtuales. 2. El administrador selecciona una serie de parámetros: - Rango de máquinas virtuales. - Número de copias de seguridad de las máquinas virtuales que pueden permanecer almacenadas. - Número de copias de seguridad de los ficheros de configuración de las máquinas virtuales que pueden permanecer almacenadas. - Número de días que las copias de seguridad obsoletas, de máquinas virtuales que ya no existen, pueden permanecer almacenadas. 3. El administrador efectúa la operación de

Identificador: CU 4	
	backup completo sobre el conjunto de máquinas virtuales. 4. Terminar.
Escenarios alternativos:	3a. El administrador escoge un rango de máquinas virtuales erróneo. 1. La acción de copia de seguridad de las máquinas virtuales termina con error para las máquinas virtuales inexistentes pero se completa para el resto de máquinas virtuales, para la limpieza de copias de seguridad obsoletas de máquinas virtuales antiguas y para la copia de los ficheros de configuración de las máquinas virtuales. 2. Volver al paso 1. 3b. El administrador no introduce un número entero como número de copias de seguridad de las máquinas virtuales que pueden permanecer guardadas. 1. La copia de seguridad de las máquinas virtuales termina con error pero se completa la copia de seguridad de los ficheros de configuración de las máquinas virtuales y la limpieza de copias de seguridad obsoletas de máquinas virtuales antiguas. 2. Volver al paso 1. 3c. El administrador no introduce un número entero como número de copias de seguridad de los ficheros de configuración de las máquinas virtuales que pueden permanecer guardadas. 1. La copia de seguridad de los ficheros de configuración de las máquinas virtuales termina con error pero se completa la copia de seguridad de las máquinas virtuales y la limpieza de copias de seguridad obsoletas de máquinas virtuales antiguas. 2. Volver al paso 1. 3d. El administrador no introduce un número entero como número de días que las copias de seguridad obsoletas, de máquinas virtuales que ya no existen,

Identificador: CU 4	
	deben permanecer almacenadas. 1. La copia de seguridad, tanto de máquinas virtuales como de ficheros de configuración de máquinas virtuales se completa, pero la limpieza de copias de seguridad obsoletas de máquinas antiguas termina con error. 2. Volver al paso 1.

Tabla 4: Definición del caso de uso CU 4

Identificador: CU 5	
Nombre:	Backup de rango de VM.
Actores:	Administrador.
Objetivo:	Se realiza una copia de seguridad de todas las máquinas virtuales de un rango establecido, eliminando las copias de seguridad de máquinas antiguas que superen un cierto número de días de antigüedad.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha realizado una copia de seguridad de cada una de las máquinas de un rango concreto y se ha comprobado, y eliminado si procedía, las copias de seguridad de máquinas virtuales antiguas.
Escenario básico:	1. El administrador accede a la pantalla de administración de copias de seguridad de máquinas virtuales. 2. El administrador selecciona una serie de parámetros: - Rango de máquinas virtuales. - Número de copias de seguridad de las máquinas virtuales que pueden permanecer almacenadas. - Número de copias de seguridad de los ficheros de configuración de las

Identificador: CU 5	
	máquinas virtuales que pueden permanecer almacenadas. 3. El administrador efectúa la operación de backup de un rango de máquinas virtuales. 4. Terminar.
Escenarios alternativos:	3a. El administrador escoge un rango de máquinas virtuales erróneo. 1. La acción de copia de seguridad de las máquinas virtuales termina con error para las máquinas virtuales inexistentes pero se completa para el resto de máquinas y para la limpieza de copias de seguridad obsoletas de máquinas virtuales antiguas. 2. Volver al paso 1. 3b. El administrador no introduce un número entero como número de copias de seguridad de las máquinas virtuales que pueden permanecer guardadas. 1. La acción de copia de seguridad de las máquinas virtuales termina con error pero se completa para la limpieza de copias de seguridad obsoletas de máquinas virtuales antiguas. 2. Volver al paso 1. 3c. El administrador no introduce un número entero como número de días que las copias de seguridad obsoletas, de máquinas virtuales que ya no existen, deben permanecer almacenadas. 1. La copia de seguridad de máquinas virtuales se completa, pero la limpieza de copias de seguridad obsoletas de máquinas antiguas termina con error. 2. Volver al paso 1.

Tabla 5: Definición del caso de uso CU 5

Identificador: CU 6	
Nombre:	Backup cruzado.
Actores:	Administrador.
Objetivo:	Se realiza una copia de los ficheros de configuración de las máquinas virtuales de un nodo en el otro y viceversa.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha realizado una copia cruzada de los ficheros de configuración de las máquinas virtuales seleccionadas en cada nodo.
Escenario básico:	1. El administrador accede a la pantalla de administración de copias de seguridad de máquinas virtuales. 2. El administrador selecciona el número de copias de seguridad de los ficheros de configuración de las máquinas virtuales que pueden permanecer almacenadas. 3. El administrador efectúa la operación de backup cruzado sobre todas las máquinas virtuales. 4. Terminar.
Escenarios alternativos:	3a. El administrador no introduce un número entero como número de copias de seguridad de los ficheros de configuración de las máquinas virtuales que pueden permanecer guardadas. 1. La acción termina con error. 2. Volver al paso 1.

Tabla 6: Definición del caso de uso CU 6

Identificador: CU 7	
Nombre:	Restaurar backup.
Actores:	Administrador.

Identificador: CU 7	
Objetivo:	Se restaura una copia de seguridad de una fecha concreta y para una máquina virtual concreta.
Precondiciones:	El administrador tiene acceso a los servidores.
Postcondiciones:	Se ha restaurado una copia de seguridad de una máquina virtual.
Escenario básico:	1. El administrador accede a la pantalla de administración de copias de seguridad de máquinas virtuales. 2. El administrador selecciona una máquina virtual. 3. El administrador accede a la pantalla de copias de seguridad disponibles de la máquina virtual seleccionada. 4. El administrador selecciona una copia de seguridad a restaurar. 5. El administrador efectúa la restauración de la copia de seguridad. 6. Terminar.
Escenarios alternativos:	3a. El administrador escoge una máquina virtual errónea. 1. La acción termina con error. 2. Volver al paso 1. 5a. El administrador escoge una copia de seguridad inexistente para la máquina virtual seleccionada. 1. La acción termina con error. 2. Volver al paso 3.

Tabla 7: Definición del caso de uso CU 7

Identificador: CU 8	
Nombre:	Recuperar VMs ante fallo de nodo.
Actores:	Administrador.

Identificador: CU 8	
Objetivo:	Se levantan todas las máquinas virtuales de un nodo caído en otro nodo activo.
Precondiciones:	El administrador tiene acceso a los servidores. El servicio de almacenamiento en red está disponible. Al menos un nodo está habilitado.
Postcondiciones:	Se levantan todas las máquinas virtuales del nodo caído en un nodo activo.
Escenario básico:	1. El administrador accede a la pantalla de copias de seguridad de los ficheros de configuración de máquinas virtuales del nodo caído. 2. El administrador efectúa la operación de restauración de todos los ficheros de configuración de las máquinas virtuales en el nodo actual. 3. El administrador enciende todas las máquinas virtuales restauradas. 4. Terminar.
Escenarios alternativos:	2a. No existen copias de seguridad de los ficheros de configuración de las máquinas virtuales. 1. Volver al paso 1.

Tabla 8: Definición del caso de uso CU 8

Identificador: CU 9	
Nombre:	Backup corporativo
Actores:	UC3M
Objetivo:	Se realiza una copia de seguridad de todas las máquinas virtuales en un almacenamiento corporativo.
Precondiciones:	La UC3M tiene acceso a los servidores.

Identificador: CU 9	
Postcondiciones:	Se ha realizado una copia de seguridad de todas las máquinas virtuales en el almacenamiento corporativo.
Escenario básico:	1. La UC3M accede a los servidores de máquinas virtuales. 2. La UC3M efectúa la operación de backup corporativo sobre todas las máquinas virtuales. 3. Terminar.
Escenarios alternativos:	2a. No existen máquinas virtuales en los servidores. 1. Volver al paso 1.

Tabla 9: Definición del caso de uso CU 9

4.2.1.2. Casos de uso del servicio de monitorización

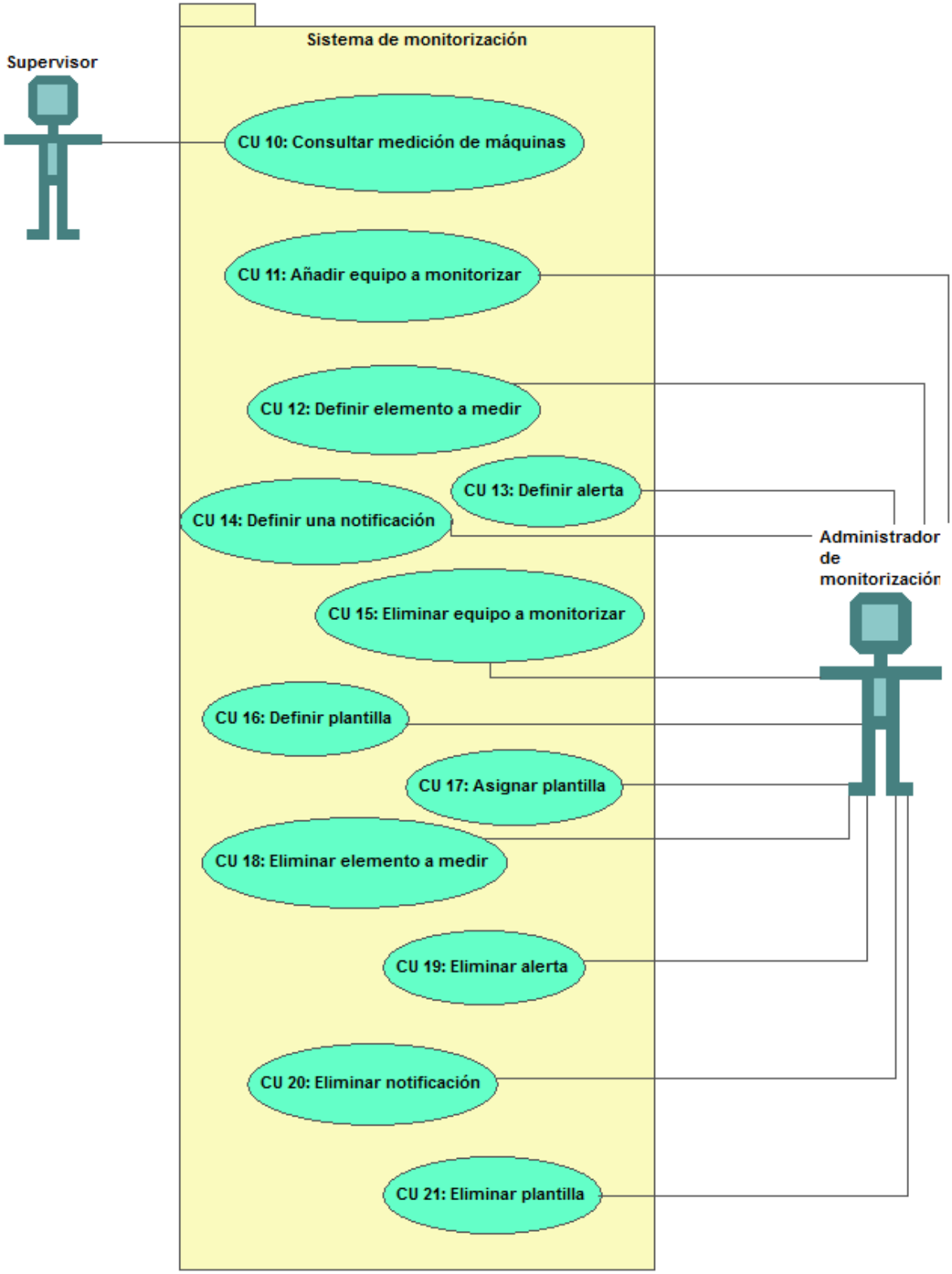

Ilustración 45 Diagrama de casos de uso para el servicio de monitorización

En la *Ilustración 45 Diagrama de casos de uso para el servicio de monitorización* se pueden observar doce nuevos casos de uso pertenecientes, en este caso, al servicio de monitorización. Ahora aparecen dos nuevos actores: el **supervisor** que se encarga de verificar el estado de las máquinas virtuales para así poder reaccionar a tiempo en caso de producirse algún problema, típicamente, este actor podría corresponderse con el **administrador** del Servicio de Teletrabajo, teniendo la capacidad de gestionar el Servicio de Teletrabajo para solventar las posibles incidencias. Por otro lado está el actor que es el encargado de mantener el sistema de monitorización, indicando y configurando los parámetros a medir de cada una de las máquinas, además de darlas de alta en el sistema. Respecto al caso de uso número 10, es un resumen de representa a otros seis casos de usos que se especificarán más adelante.

Identificador: CU 10	
Nombre:	Consultar medición de máquinas
Actores:	Supervisor
Objetivo:	Conocer el estado de funcionamiento de las máquinas de la red.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo. 2. El supervisor accede a la sección de consultar máquinas. 3. El supervisor selecciona un grupo de máquinas. 4. El supervisor selecciona la máquina deseada de un grupo. 5. El supervisor selecciona el parámetro medido que desea consultar. 6. El supervisor obtiene los datos deseados.
Escenarios	3a. No hay grupos definidos.

Identificador: CU 10	
alternativos:	1. Volver al paso 2. 4a. No hay máquinas en el grupo. 1. Volver al paso 3. 5a. No se realizan mediciones sobre la máquina. 1. Volver al paso 4.

Tabla 10: Definición del caso de uso CU 10

Identificador: CU 11	
Nombre:	Añadir equipo a monitorizar
Actores:	Administrador de monitorización
Objetivo:	Comenzar la monitorización de una nueva máquina.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y la máquina a monitorizar está preparada para ser monitorizada.
Postcondiciones:	La máquina nueva está siendo monitorizada.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona la opción “añadir máquina”. 4. El administrador de monitorización añade la dirección de red de la nueva máquina. 5. El administrador selecciona el grupo al que asignar la máquina. 6. La máquina aparece en la interfaz como monitorizada.
Escenarios alternativos:	5a. No existe el grupo al que pertenece la máquina. 1. El administrador de monitorización selecciona la opción “nuevo grupo”.

Identificador: CU 11	
	2. El administrador de monitorización escribe el nombre del nuevo grupo en el cuadro de texto que se le presenta. 3. El administrador de monitorización acepta el nuevo nombre de grupo. 4. Saltar al paso 6. 6a. La máquina no aparece como monitorizada. 1. Terminar con error. 2. Volver al paso 3.

Tabla 11: Definición del caso de uso CU 11

Identificador: CU 12	
Nombre:	Definir elemento a medir
Actores:	Administrador de monitorización
Objetivo:	Comenzar a medir y almacenar un dato concreto sobre una máquina.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y se está monitorizando al menos una máquina.
Postcondiciones:	Se monitoriza un nuevo parámetro de la máquina deseada.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona el grupo de la máquina deseada. 4. El administrador de monitorización escoge la máquina del grupo a la que añadir el medidor. 5. El administrador de monitorización

Identificador: CU 12	
	selecciona la opción “elementos medidos”. 6. El administrador de monitorización escoge la opción “nuevo elemento”. 7. El administrador de monitorización escoge de la lista el elemento que desea medir. 8. El administrador de monitorización escribe la frecuencia con la que se medirá el elemento. 9. El administrador de monitorización escribe el número de días que se almacenara un valor de ese elemento. 10. El administrador de monitorización acepta el nuevo elemento. 11. El nuevo elemento aparece con el estado “activado”.
Escenarios alternativos:	4a. El administrador de monitorización desea añadir el elemento a monitorizar a una plantilla. 1. El administrador de monitorización escoge la plantilla del grupo a la que añadir el medidor. 2. El administrador de monitorización selecciona la opción “elementos medidos”. 3. El administrador de monitorización escoge la opción “nuevo elemento”. 4. El administrador de monitorización escoge de la lista el elemento que desea medir. 5. El administrador de monitorización escribe la frecuencia con la que se medirá el elemento. 6. El administrador de monitorización escribe el número de días que se almacenara un valor de ese elemento. 7. El administrador de monitorización acepta el nuevo elemento. 4b. El grupo no contiene máquinas. 1. Vuelve al paso 3. 7a. La lista no ofrece el elemento a medir.

Identificador: CU 12	
	1. El administrador de monitorización escribe con una sintaxis definida la especificación del nuevo elemento a seguir en el cuadro de texto que se le muestra. 2. Saltar al paso 8. 8a. La frecuencia introducida no presenta un formato numérico válido. 1. Error al añadir elemento. 2. Volver al paso 8. 9a. La fecha de caducidad del elemento no presenta un formato numérico válido. 1. Error al añadir elemento. 2. Volver al paso 9. 10a. El administrador de monitorización cancela el nuevo elemento. 1. Volver al paso 5. 11a. El elemento añadido aparece como "desactivado". 1. Error, el elemento no puede ser medido o está mal definido. 2. Volver al paso 5.

Tabla 12: Definición del caso de uso CU 12

Identificador: CU 13	
Nombre:	Definir alerta
Actores:	Administrador de monitorización
Objetivo:	Crear un disparador que mida un elemento y de una notificación en la interfaz cuando se cumpla cierta condición.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y está definido al menos un elemento a monitorizar.

Identificador: CU 13	
Postcondiciones:	La máquina deseada esta supervisada por un nuevo disparador.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona el grupo de la máquina deseada. 4. El administrador de monitorización escoge la máquina del grupo a la que añadir el disparador. 5. El administrador de monitorización escoge la opción “disparadores”. 6. El administrador de monitorización escoge la opción “añadir disparador”. 7. El administrador de monitorización escoge un elemento monitorizado de la máquina. 8. El administrador de monitorización selecciona el operador lógico a aplicar. 9. El administrador de monitorización escribe el valor con el que quiere comparar el elemento. 10. El administrador de monitorización acepta la condición introducida. 11. El administrador de monitorización escoge el nivel de alerta del disparador. 12. El administrador de monitorización acepta el disparador. 13. El disparador aparece como “activado” en la lista de disparadores.
Escenarios alternativos:	4a. El administrador de monitorización desea añadir el disparador a una plantilla. 1. El administrador de monitorización escoge la plantilla del grupo a la que añadir el disparador. 2. El administrador de monitorización escoge la

Identificador: CU 13	
	opción “disparadores”. 3. El administrador de monitorización escoge la opción “añadir disparador”. 4. El administrador de monitorización escoge un elemento monitorizado de la máquina. 5. El administrador de monitorización selecciona el operador lógico a aplicar. 6. El administrador de monitorización escribe el valor con el que quiere comparar el elemento. 7. El administrador de monitorización acepta la condición introducida. 8. El administrador de monitorización escoge el nivel de alerta del disparador. 9. El administrador de monitorización acepta el disparador. 4b. El grupo no contiene máquinas. 1. Vuelve al paso 3. 7a. La máquina no está monitorizando ningún elemento. 1. Pulsar la opción “nuevo elemento”. 2. Seguir los pasos del caso de uso CU 12. 3. Saltar al paso 8. 8a. El operador seleccionado no es aplicable al elemento monitorizado. 1. Error al añadir disparador. 2. Volver al paso 8. 9a. El valor introducido no es del mismo tipo de datos devuelto por el elemento monitorizado. 1. Error al añadir disparador. 2. Volver al paso 9. 10a. El administrador de monitorización desea añadir una nueva condición al disparador. 1. El administrador de monitorización acepta la condición introducida.

Identificador: CU 13	
	2. El administrador de monitorización selecciona la opción “nueva condición”. 3. El administrador de monitorización escoge el operador booleano que se aplicara entre ambas condiciones. 4. Volver al paso 7. 12a. El administrador de monitorización cancela el nuevo disparador. 1. Volver al paso 6. 13a. El administrador de monitorización no desea activar el disparador. 1. El administrador de monitorización desactiva la casilla “activado” del disparador.

Tabla 13: Definición del caso de uso CU 13

Identificador: CU 14	
Nombre:	Definir una notificación
Actores:	Administrador de monitorización
Objetivo:	Hacer que el sistema envíe alertas al supervisor sin necesidad de que acceda a la interfaz.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y está definida al menos una alerta.
Postcondiciones:	Queda definida una notificación asociada a una alerta.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar notificaciones. 3. El administrador de monitorización selecciona la opción “nueva notificación”. 4. El administrador de monitorización

Identificador: CU 14	
	selecciona la opción “añadir condición”. 5. El administrador de monitorización selecciona un grupo. 6. El administrador de monitorización selecciona una máquina del grupo. 7. El administrador de monitorización selecciona una alerta de la máquina. 8. El administrador de monitorización selecciona el valor de la alerta que quiere que se notifique. 9. El administrador de monitorización acepta la condición. 10. El administrador de monitorización introduce la dirección de correo electrónico a la que se mandará la notificación. 11. El administrador de monitorización acepta la nueva notificación. 12. En la interfaz se muestra la nueva notificación en estado “activado”.
Escenarios alternativos:	6a. El grupo no contiene máquinas. 1. Volver al paso 5. 6b. El administrador de monitorización desea añadir la notificación a una plantilla. 1. El administrador de monitorización selecciona la plantilla. 2. El administrador de monitorización selecciona una alerta definida en la plantilla. 3. Saltar al paso 8. 7a. La máquina no tiene alertas definidas. 1. Volver al paso 6. 9a. El administrador de monitorización desea añadir una nueva condición. 1. El administrador de monitorización acepta la condición anterior. 2. El administrador de monitorización

Identificador: CU 14	
	selecciona el operador booleano que se aplicará entre las condiciones. 3. Volver al paso 4. 11a. El administrador de monitorización cancela la nueva notificación. 1. Volver al paso 3. 12a. El administrador de monitorización no desea que la nueva notificación esté activada. 1. El administrador de monitorización desactiva la casilla "activado" de la notificación.

Tabla 14: Definición del caso de uso CU 14

Identificador: CU 15	
Nombre:	Eliminar equipo a monitorizar
Actores:	Administrador de monitorización
Objetivo:	Dejar de monitorizar un equipo.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y se está monitorizando al menos una máquina.
Postcondiciones:	Una máquina deja de ser monitorizada.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona el grupo de la máquina deseada. 4. El administrador de monitorización selecciona una máquina del grupo. 5. El administrador de monitorización selecciona la opción "eliminar". 6. La máquina ha desaparecido del grupo.

Identificador: CU 15	
Escenarios alternativos:	4a. El grupo seleccionado no tiene ninguna máquina. 1. Volver al paso 3. 5a. El administrador de monitorización selecciona la opción “desactivar”. 1. La máquina aparece como no monitorizada. 2. Terminar.

Tabla 15: Definición del caso de uso CU 15

Identificador: CU 16	
Nombre:	Definir plantilla
Actores:	Administrador de monitorización
Objetivo:	Crear una plantilla que automatice la configuración de las nuevas máquinas a monitorizar.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización.
Postcondiciones:	El sistema tiene definido una nueva plantilla.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar plantillas. 3. El administrador de monitorización selecciona la opción “nueva plantilla”. 4. El administrador de monitorización selecciona el grupo en el que se incluirá la plantilla. 5. El administrador de monitorización introduce el nombre de la plantilla. 6. El administrador de monitorización acepta la nueva plantilla. 7. Aparece la nueva plantilla en la lista de plantillas y en la lista de máquinas del grupo

Identificador: CU 16	
	seleccionado.
Escenarios alternativos:	3a. El administrador de monitorización selecciona una plantilla ya existente. 1. El administrador de monitorización selecciona la opción “copiar plantilla”. 2. Saltar a paso 4. 3b. El administrador de monitorización selecciona una plantilla ya existente. 1. El administrador de monitorización selecciona la opción “heredar de plantilla”. 2. Saltar a paso 4. 4a. No existe el grupo al que pertenece la plantilla. 1. El administrador de monitorización selecciona la opción “nuevo grupo”. 2. El administrador de monitorización escribe el nombre del nuevo grupo en el cuadro de texto que se le presenta. 3. El administrador de monitorización acepta el nuevo nombre de grupo. 4. Saltar al paso 5. 6a. El administrador de monitorización cancela la nueva plantilla. 1. Volver al paso 3.

Tabla 16: Definición del caso de uso CU 16

Identificador: CU 17	
Nombre:	Asignar plantilla
Actores:	Administrador de monitorización
Objetivo:	Añadir a una máquina monitorizada los elementos de monitorización, alertas y notificaciones de una plantilla.

Identificador: CU 17	
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización, está monitorizada al menos una máquina y está definida al menos una plantilla.
Postcondiciones:	La máquina modificada adquiere todos los elementos de monitorización, alertas y notificaciones de la plantilla.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona un grupo. 4. El administrador de monitorización selecciona una máquina del grupo. 5. El administrador de monitorización selecciona la opción “editar máquina”. 6. El administrador de monitorización selecciona la opción “añadir plantilla”. 7. El administrador de monitorización selecciona la plantilla a aplicar de la lista. 8. El administrador de monitorización acepta la plantilla seleccionada. 9. El administrador de monitorización acepta los cambios en la máquina.
Escenarios alternativos:	4a. El grupo seleccionado no contiene máquinas. 1. Volver al paso 3. 7a. La lista de plantillas está vacía. 1. Error al aplicar plantilla. 2. Volver al paso 6. 8a. El administrador de monitorización cancela la plantilla seleccionada. 1. Volver a 6. 9a. El administrador de monitorización cancela los

Identificador: CU 17	
	cambios en la máquina. 1. Volver a 4.

Tabla 17: Definición del caso de uso CU 17

Identificador: CU 18	
Nombre:	Eliminar elemento a medir
Actores:	Administrador de monitorización
Objetivo:	Dejar de monitorizar un determinado elemento de una máquina.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y existe al menos un elemento monitorizado.
Postcondiciones:	Se deja de monitorizar un elemento.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona un grupo. 4. El administrador de monitorización selecciona una máquina del grupo. 5. El administrador de monitorización selecciona la opción “elementos medidos”. 6. El administrador de monitorización escoge el elemento que desea eliminar. 7. El administrador de monitorización selecciona la opción “eliminar”. 8. El elemento monitorizado ha desaparecido de la lista de elementos.
Escenarios alternativos:	4a. El grupo seleccionado no contiene máquinas. 1. Volver al paso 3.

Identificador: CU 18	
	4b. El administrador de monitorización escoge una plantilla del grupo. 1. Saltar al paso 5. 6a. La máquina seleccionada no contiene elementos monitorizados. 1. Volver al paso 4. 7a. El administrador de monitorización desactiva la casilla “activado”. 1. Terminar. 8a. El elemento a eliminar proviene de una plantilla. 1. Falla eliminar elemento. 2. Terminar.

Tabla 18: Definición del caso de uso CU 18

Identificador: CU 19	
Nombre:	Eliminar alerta
Actores:	Administrador de monitorización
Objetivo:	Dejar de obtener avisos de alerta de una máquina.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y existe al menos un disparador.
Postcondiciones:	Se deja de supervisar un elemento mediante un disparador.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar máquinas. 3. El administrador de monitorización selecciona un grupo. 4. El administrador de monitorización selecciona una máquina del grupo.

Identificador: CU 19	
	5. El administrador de monitorización selecciona la opción “disparadores”. 6. El administrador de monitorización escoge el disparador que desea eliminar. 7. El administrador de monitorización selecciona la opción “eliminar”. 8. El disparador monitorizado ha desaparecido de la lista de elementos.
Escenarios alternativos:	4a. El grupo seleccionado no contiene máquinas. 1. Volver al paso 3. 4b. El administrador de monitorización escoge una plantilla del grupo. 1. Saltar al paso 5. 6a. La máquina seleccionada no contiene disparadores. 1. Volver al paso 4. 7a. El administrador de monitorización desactiva la casilla “activado”. 1. El administrador de monitorización acepta el cambio. 2. Terminar. 8a. El disparador a eliminar proviene de una plantilla. 1. Falla eliminar disparador. 2. Terminar.

Tabla 19: Definición del caso de uso CU 19

Identificador: CU 20	
Nombre:	Eliminar notificación
Actores:	Administrador de monitorización
Objetivo:	Dejar de recibir notificaciones por correo electrónico

Identificador: CU 20	
	referentes a una determinada alerta.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y existe al menos una notificación.
Postcondiciones:	El sistema queda configurado para dejar de emitir notificaciones cuando se active una determinada alerta.
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar notificaciones. 3. El administrador de monitorización selecciona la notificación a eliminar. 4. El administrador de monitorización selecciona la opción “eliminar”. 5. La notificación eliminada ha desaparecido de la lista de notificaciones.
Escenarios alternativos:	4a. El administrador de monitorización desactiva la casilla “activado”. 1. El administrador de monitorización acepta el cambio. 2. Terminar.

Tabla 20: Definición del caso de uso CU 20

Identificador: CU 21	
Nombre:	Eliminar plantilla
Actores:	Administrador de monitorización
Objetivo:	Eliminar una plantilla de configuración del sistema.
Precondiciones:	El administrador de monitorización tiene acceso al servidor de monitorización y existe al menos una plantilla.

Identificador: CU 21	
Postcondiciones:	Desaparece la plantilla deseada del sistema,
Escenario básico:	1. El administrador de monitorización accede a la interfaz del servicio de monitorización. 2. El administrador de monitorización accede a la sección de gestionar plantillas. 3. El administrador de monitorización escoge la plantilla deseada. 4. El administrador de monitorización selecciona la opción “eliminar”. 5. La plantilla eliminada ha desaparecido de la lista de plantillas.
Escenarios alternativos:	2a. El administrador de monitorización accede a la sección de gestionar de máquinas. 1. El administrador de monitorización selecciona un grupo. 2. El administrador de monitorización selecciona la plantilla a eliminar del grupo. 3. El administrador de monitorización selecciona la opción “eliminar”. 4. La plantilla eliminada ha desaparecido del grupo.

Tabla 21: Definición del caso de uso CU 21

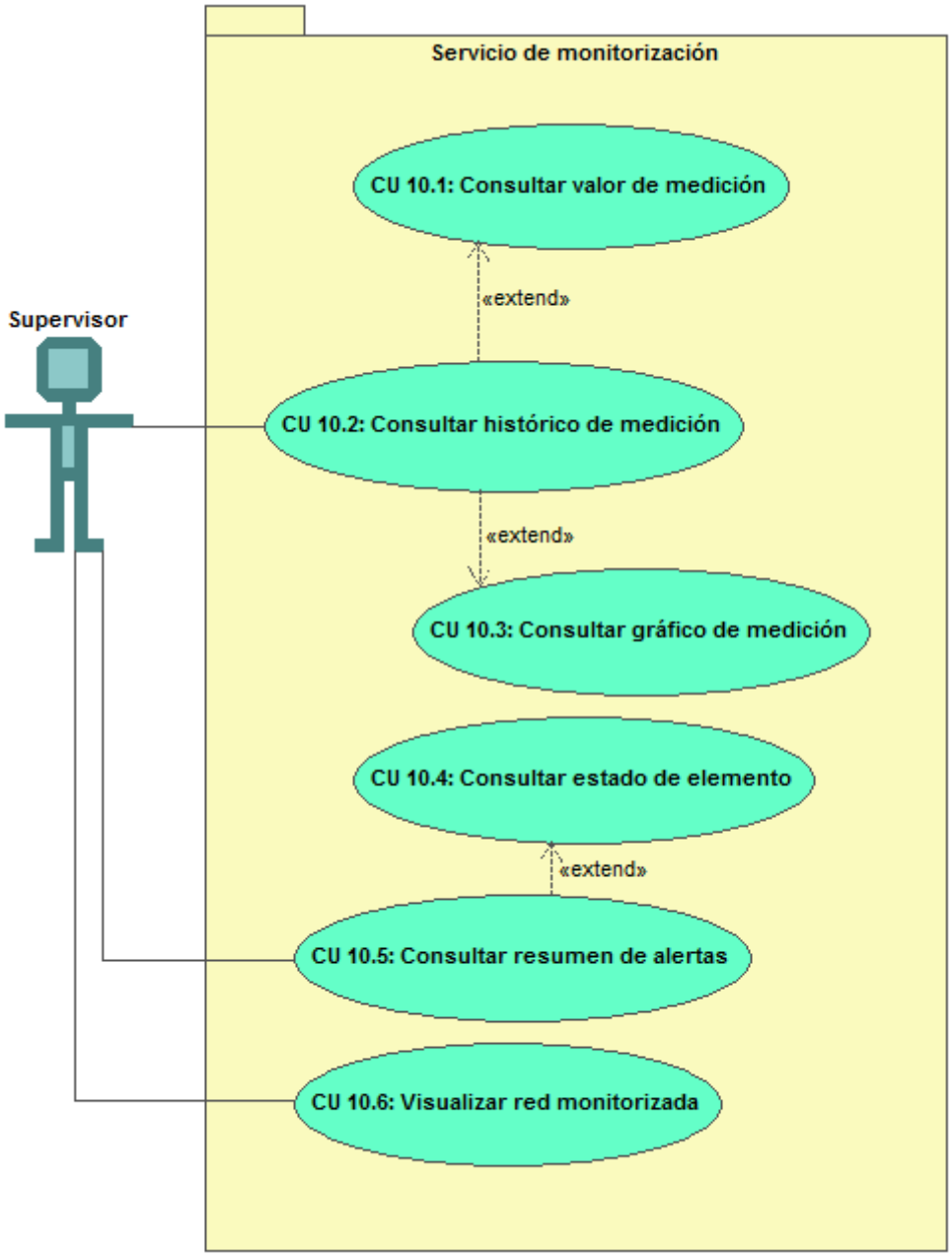

Ilustración 46 Diagrama de casos de uso para el supervisor

El último diagrama de casos de uso, muestra el despliegue del caso de uso CU 10 con todas las acciones (seis consultas en este caso) que puede realizar el actor **supervisor** con el servicio de monitorización.

Identificador: CU 10.1	
Nombre:	Consultar valor de medición
Actores:	Supervisor
Objetivo:	Conocer el valor actual de un elemento monitorizado en una máquina.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y existe un elemento monitorizado.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo. 2. El supervisor accede a la sección de consultar máquinas. 3. El supervisor selecciona un grupo de máquinas. 4. El supervisor selecciona la opción “mostrar datos”. 5. El sistema muestra el valor de todos los elementos monitorizados de las máquinas de ese grupo.
Escenarios alternativos:	4a. No hay máquinas en el grupo. 1. Volver al paso 3.

Tabla 22: Definición del caso de uso CU 10.1

Identificador: CU 10.2	
Nombre:	Consultar histórico de medición
Actores:	Supervisor
Objetivo:	Conocer los últimos valores de un elemento monitorizado en una máquina.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y existe un elemento monitorizado.

Identificador: CU 10.2	
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo. 2. El supervisor accede a la sección de consultar máquinas. 3. El supervisor selecciona un grupo de máquinas. 4. El supervisor selecciona la opción “mostrar datos”. 5. El supervisor pincha sobre el elemento monitorizado cuyos datos quiere conocer. 6. El supervisor selecciona la opción “ver últimos valores”. 7. El sistema muestra una tabla con todos los valores almacenados en la base de datos del elemento monitorizado.
Escenarios alternativos:	4a. No hay máquinas en el grupo. 2. Volver al paso 3.

Tabla 23: Definición del caso de uso CU 10.2

Identificador: CU 10.3	
Nombre:	Consultar gráfico de medición
Actores:	Supervisor
Objetivo:	Conocer los últimos valores de un elemento monitorizado en una máquina de forma gráfica.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y existe un elemento monitorizado.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo.

Identificador: CU 10.3	
	2. El supervisor accede a la sección de consultar máquinas. 3. El supervisor selecciona un grupo de máquinas. 4. El supervisor selecciona la opción “mostrar datos”. 5. El supervisor pincha sobre el elemento monitorizado cuyos datos quiere conocer. 6. El supervisor selecciona la opción “ver gráfico”. 7. El supervisor selecciona el espacio de tiempo cuyos datos quiera conocer. 8. El sistema muestra un gráfico con inicio y fin en el espacio temporal seleccionado.
Escenarios alternativos:	4a. No hay máquinas en el grupo. 1. Volver al paso 3. 8a. No hay datos para el intervalo de tiempo seleccionado. 1. Volver al paso 7.

Tabla 24: Definición del caso de uso CU 10.3

Identificador: CU 10.4	
Nombre:	Consultar estado de elemento
Actores:	Supervisor
Objetivo:	Conocer si un elemento monitorizado está funcionando correctamente.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y existe una alerta.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo.

Identificador: CU 10.4	
	2. El supervisor accede a la sección de consultar máquinas. 3. El supervisor selecciona un grupo de máquinas. 4. El supervisor selecciona la máquina deseada de un grupo. 5. El sistema muestra el estado de todos los disparadores que hay definidos para la máquina.
Escenarios alternativos:	4a. No hay máquinas en el grupo. 1. Volver al paso 3. 5a. No se realizan mediciones sobre la máquina. 1. Volver al paso 4.

Tabla 25: Definición del caso de uso CU 10.4

Identificador: CU 10.5	
Nombre:	Consultar resumen de alertas
Actores:	Supervisor
Objetivo:	Conocer el estado de todas las máquinas de la red monitorizada.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización y existe una alerta.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo. 2. El supervisor accede a la sección resumen. 3. El sistema muestra un resumen del estado de todas las alertas.

Tabla 26: Definición del caso de uso CU 10.5

Identificador: CU 10.6	
Nombre:	Visualizar red monitorizada
Actores:	Supervisor
Objetivo:	Visualizar gráficamente el estado de las máquinas monitorizadas y sus posiciones y conexiones en la red.
Precondiciones:	El supervisor tiene acceso al servidor de monitorización.
Postcondiciones:	-
Escenario básico:	1. El supervisor accede a la interfaz del servidor de teletrabajo. 2. El supervisor accede a la sección de pantallas. 3. El sistema muestra todas las subredes monitorizadas y si hay algún problema con al menos una máquina de la subred. 4. El supervisor pincha sobre la subred que desea visualizar. 5. El sistema muestra todos los equipos de esa subred y un resumen del estado de cada máquina.
Escenarios alternativos:	3a. No hay red dibujada. 1. Error al mostrar red. 2. Terminar 4a. No hay subredes definidas. 1. Se muestra directamente el resumen de los equipos del sistema. 2. Terminar.

Tabla 27: Definición del caso de uso CU 10.6

4.2.2. Especificación de requisitos

Una vez definidos los casos de uso, y conociendo que es lo que se quiere que haga el sistema, nos podemos apoyar en ellos para iniciar un análisis en mayor profundidad especificando cuáles son los requisitos software de los sistemas.

La especificación de requisitos se dividirá en dos tipos, funcionales y no funcionales, y contendrá y continuará la especificación de requisitos del proyecto previo (Gil Bázquez, 2011) para dotar de contexto a los nuevos requisitos. Los requisitos que no se desarrollarán durante este proyecto, porque ya han sido incluidos en el proyecto anterior, se presentarán en tablas de color negro.

4.2.2.1. Requisitos Funcionales

Identificador: RSF-01	
Título:	Introducir dirección IP o nombre de máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en la que introducir la dirección IP o el nombre de la máquina virtual para conectarse a ella.

Tabla 28: Requisito del software RSF-01

Identificador: RSF-02	
Título:	Introducir datos de inicio de sesión.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en la que introducir los datos de acceso (nombre de usuario y

Identificador: RSF-02	
	contraseña) a la máquina virtual.

Tabla 29: Requisito del software RSF-02

Identificador: RSF-03	
Título:	Pantalla de administración de máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla de administración de máquinas virtuales.

Tabla 30: Requisito del software RSF-03

Identificador: RSF-04	
Título:	Opción de crear una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para crear una máquina virtual.

Tabla 31: Requisito del software RSF-04

Identificador: RSF-05	
Título:	Pantalla de creación de máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.

Identificador: RSF-05	
Descripción:	Se deberá proporcionar una pantalla de creación de máquinas virtuales que permita insertar los siguientes parámetros: - Nombre a asignar a la máquina virtual. - Cantidad de memoria RAM a asignar a la máquina virtual. - Número de procesadores a asignar a la máquina virtual. - Tamaño de disco duro a asignar a la máquina virtual. - Tipo de almacenamiento del disco duro. - Tipo de tarjeta de red a asignar a la máquina virtual. - Nodo en el que almacenar la máquina virtual. - Medio a utilizar para la instalación del sistema operativo.

Tabla 32: Requisito del software RSF-05

Identificador: RSF-06	
Título:	Opción de modificar una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para modificar una máquina virtual.

Tabla 33: Requisito del software RSF-06

Identificador: RSF-07	
Título:	Pantalla de configuración de máquinas virtuales.
Tipo:	Funcional.

Identificador: RSF-07	
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla de configuración de máquinas virtuales que permita configurar los siguientes parámetros: - Nombre de la máquina virtual. - Cantidad de memoria RAM de la máquina virtual. - Número de procesadores de la máquina virtual. - Medio a utilizar para la instalación del sistema operativo. - Orden de arranque de dispositivos de la máquina virtual.

Tabla 34: Requisito del software RSF-07

Identificador: RSF-08	
Título:	Opción de añadir un disco duro a una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para añadir un disco duro a una máquina virtual desde la pantalla de configuración de dicha máquina virtual.

Tabla 35: Requisito del software RSF-08

Identificador: RSF-09	
Título:	Pantalla de inserción de disco duro a una máquina virtual.
Tipo:	Funcional.

Identificador: RSF-09	
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de inserción de disco duro para una máquina virtual que permita configurar los siguientes parámetros: - Tamaño del disco duro. - Tipo de almacenamiento.

Tabla 36: Requisito del software RSF-09

Identificador: RSF-10	
Título:	Opción de eliminar un disco duro de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para eliminar un disco duro de una máquina virtual.

Tabla 37: Requisito del software RSF-10

Identificador: RSF-11	
Título:	Opción de añadir una tarjeta de red a una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para añadir una tarjeta de red a una máquina virtual.

Tabla 38: Requisito del software RSF-11

Identificador: RSF-12	
Título:	Pantalla de inserción de tarjeta de red a una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de inserción de tarjeta de red para una máquina virtual que permita indicar el tipo de tarjeta de red.

Tabla 39: Requisito del software RSF-12

Identificador: RSF-13	
Título:	Opción de eliminar una tarjeta de red de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para eliminar una tarjeta de red de una máquina virtual.

Tabla 40: Requisito del software RSF-13

Identificador: RSF-14	
Título:	Opción de añadir un dispositivo extraíble a una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para añadir un dispositivo extraíble a una máquina virtual.

Tabla 41: Requisito del software RSF-14

Identificador: RSF-15	
Título:	Pantalla de inserción de dispositivo extraíble a una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de inserción de dispositivo extraíble para una máquina virtual que permita indicar el medio a montar en éste.

Tabla 42: Requisito del software RSF-15

Identificador: RSF-16	
Título:	Opción de eliminar un dispositivo extraíble de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para eliminar un dispositivo extraíble de una máquina virtual.

Tabla 43: Requisito del software RSF-16

Identificador: RSF-17	
Título:	Opción de eliminar una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para eliminar una máquina virtual.

Tabla 44: Requisito del software RSF-17

Identificador: RSF-18	
Título:	Opción de clonar una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para clonar una máquina virtual a partir de otra.

Tabla 45: Requisito del software RSF-18

Identificador: RSF-19	
Título:	Opción de migrar una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para migrar una máquina virtual.

Tabla 46: Requisito del software RSF-19

Identificador: RSF-20	
Título:	Pantalla de migración de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de migración de una máquina virtual que permita indicar el nodo de destino.

Tabla 47: Requisito del software RSF-20

Identificador: RSF-21	
Título:	Opción de redimensionar un disco duro de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para redimensionar un disco duro de una máquina virtual.

Tabla 48: Requisito del software RSF-21

Identificador: RSF-22	
Título:	Pantalla de redimensionado de disco duro de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de redimensionado de disco duro para una máquina virtual que permita indicar el nuevo tamaño de dicho disco duro.

Tabla 49: Requisito del software RSF-22

Identificador: RSF-23	
Título:	Opción de encender una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para encender una máquina virtual.

Tabla 50: Requisito del software RSF-23

Identificador: RSF-24	
Título:	Opción de apagar una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para apagar una máquina virtual.

Tabla 51: Requisito del software RSF-24

Identificador: RSF-25	
Título:	Opción de detener una máquina virtual.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para detener una máquina virtual.

Tabla 52: Requisito del software RSF-25

Identificador: RSF-26	
Título:	Opción de clonar varias veces una máquina virtual.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para clonar una máquina virtual a partir de otra varias veces.

Tabla 53: Requisito del software RSF-26

Identificador: RSF-27	
Título:	Pantalla de clonación múltiple de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla de clonación de una máquina virtual que permita indicar cuántas copias se quieren realizar.

Tabla 54: Requisito del software RSF-27

Identificador: RSF-28	
Título:	Opción de eliminar varias máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para eliminar varias máquinas virtuales.

Tabla 55: Requisito del software RSF-28

Identificador: RSF-29	
Título:	Opción de encender varias máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para encender varias máquinas virtuales.

Tabla 56: Requisito del software RSF-29

Identificador: RSF-30	
Título:	Opción de apagar varias máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para apagar varias máquinas virtuales.

Tabla 57: Requisito del software RSF-30

Identificador: RSF-31	
Título:	Opción de detener varias máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para detener varias máquinas virtuales.

Tabla 58: Requisito del software RSF-31

Identificador: RSF-32	
Título:	Pantalla de administración de copias de seguridad de máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla de administración de copias de seguridad de máquinas virtuales que permita insertar los siguientes parámetros: - Rango de máquinas virtuales.

Identificador: RSF-32	
	- Número de copias de seguridad de las máquinas virtuales que pueden permanecer almacenadas. - Número de copias de seguridad de los ficheros de configuración de las máquinas virtuales que pueden permanecer almacenadas. - Número de días que las copias de seguridad obsoletas, de máquinas virtuales que ya no existen, pueden permanecer almacenadas.

Tabla 59: Requisito del software RSF-32

Identificador: RSF-33	
Título:	Opción de backup completo.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para realizar un backup completo.

Tabla 60: Requisito del software RSF-33

Identificador: RSF-34	
Título:	Opción de backup de un rango de máquinas virtuales.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para realizar un backup de un rango de máquinas virtuales.

Tabla 61: Requisito del software RSF-34

Identificador: RSF-35	
Título:	Opción de backup cruzado.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para realizar un backup cruzado.

Tabla 62: Requisito del software RSF-35

Identificador: RSF-36	
Título:	Opción de listar las copias de seguridad de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para listar las copias de seguridad de una máquina virtual.

Tabla 63: Requisito del software RSF-36

Identificador: RSF-37	
Título:	Pantalla de copias de seguridad disponibles de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de copias de seguridad disponibles.

Tabla 64: Requisito del software RSF-37

Identificador: RSF-38	
Título:	Opción de restaurar una copia de seguridad de una máquina virtual.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para restaurar una copia de seguridad de una máquina virtual.

Tabla 65: Requisito del software RSF-38

Identificador: RSF-39	
Título:	Pantalla de copias de seguridad de los ficheros de configuración de máquinas virtuales de otro nodo.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporciona una pantalla de copias de seguridad de ficheros de configuración de máquinas virtuales de otro nodo disponibles.

Tabla 66: Requisito del software RSF-39

Identificador: RSF-40	
Título:	Opción de restaurar copias de seguridad de los ficheros de configuración de máquinas virtuales de otro nodo.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.

Identificador: RSF-40	
Descripción:	Se deberá proporcionar una opción para restaurar copias de seguridad de los ficheros de configuración de máquinas virtuales de otro nodo.

Tabla 67: Requisito del software RSF-40

Identificador: RSF-41	
Título:	Opción de backup corporativo.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción para realizar un backup corporativo.

Tabla 68: Requisito del software RSF-41

Identificador: RSF-42	
Título:	Pantalla de valores de datos.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en la que se muestren los valores de los elementos medidos por el sistema de monitorización.

Tabla 69: Requisito del software RSF-42

Identificador: RSF-43	
Título:	Opción de consultar histórico de datos
Tipo:	Funcional.
Prioridad:	Alta.

Identificador: RSF-43	
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción durante la consulta de los elementos monitorizados para consultar todos los valores que ha tenido ese elemento de forma numérica o textual.

Tabla 70: Requisito del software RSF-43

Identificador: RSF-44	
Título:	Opción de visualizar el histórico de datos gráficamente.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una opción durante la consulta de los elementos monitorizados para consultar todos los valores que ha tenido ese elemento en un gráfico.

Tabla 71: Requisito del software RSF-44

Identificador: RSF-45	
Título:	Pantalla de estado de los elementos.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla que muestre el estado de los disparadores asociados a un equipo monitorizado.

Tabla 72: Requisito del software RSF-45

Identificador: RSF-46	
Título:	Pantalla de resumen de la red.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla que ofrezca un resumen del estado de todos los disparadores monitorizados.

Tabla 73: Requisito del software RSF-46

Identificador: RSF-47	
Título:	Pantalla de esquema de red
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Opcional.
Estabilidad:	Media.
Descripción:	Se deberá proporcionar una pantalla que muestre esquemáticamente los elementos de la red monitorizada con sus conexiones.

Tabla 74: Requisito del software RSF-47

Identificador: RSF-48	
Título:	Opción de visualizar incidencias en el esquema de red.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Media.
Descripción:	Se deberá mostrar para cada quipo representado en el esquema de red si ese equipo tiene disparadores en un estado de alerta.

Tabla 75: Requisito del software RSF-48

Identificador: RSF-49	
Título:	Opción de añadir equipo a monitorizar.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de agregar un nuevo equipo para ser monitorizado introduciendo la dirección IP o nombre DNS del equipo. El nuevo equipo debe estar configurado para ser monitorizado.

Tabla 76: Requisito del software RSF-49

Identificador: RSF-50	
Título:	Opción de añadir elemento monitorizado.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de agregar un nuevo elemento a monitorizar en un equipo monitorizado o una plantilla.

Tabla 77: Requisito del software RSF-50

Identificador: RSF-51	
Título:	Opción de desactivar elemento monitorizado.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de desactivar un

Identificador: RSF-51	
	elemento monitorizado para que deje de ser monitorizado sin necesidad de que sea eliminado.

Tabla 78: Requisito del software RSF-51

Identificador: RSF-52	
Título:	Pantalla de elementos monitorizados.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Media.
Descripción:	Se deberá proporcionar una pantalla que muestre todos los elementos monitorizados de un equipo o plantilla y permita su gestión.

Tabla 79: Requisito del software RSF-52

Identificador: RSF-53	
Título:	Opción de activar elementos monitorizados.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de activar un elemento monitorizado previamente desactivado para que vuelva a ser monitorizado.

Tabla 80: Requisito del software RSF-53

Identificador: RSF-54	
Título:	Pantalla de gestión de máquinas.
Tipo:	Funcional.
Prioridad:	Alta.

Identificador: RSF-54	
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla que muestre todos los equipos y plantillas del sistema y permita la gestión de las mismas.

Tabla 81: Requisito del software RSF-54

Identificador: RSF-55	
Título:	Opción de crear grupo.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de crear un grupo que sirva para etiquetar un conjunto de equipos o plantillas haciendo más sencilla su gestión.

Tabla 82: Requisito del software RSF-55

Identificador: RSF-56	
Título:	Opción de agregar a grupo.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de añadir un equipo o plantilla a un grupo. Ya esté o no el equipo o plantilla asignado a otro grupo.

Tabla 83: Requisito del software RSF-56

Identificador: RSF-57	
Título:	Opción de quitar de grupo.

Identificador: RSF-57	
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de quitar un equipo o plantilla pertenecientes a un determinado grupo del mismo.

Tabla 84: Requisito del software RSF-57

Identificador: RSF-58	
Título:	Opción de eliminar grupo.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de eliminar un grupo existente del sistema, quitando todas las máquinas o plantillas que perteneciesen a ese grupo.

Tabla 85: Requisito del software RSF-58

Identificador: RSF-59	
Título:	Opción de eliminar elemento monitorizado.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de poder seleccionar un elemento monitorizado de un equipo o plantilla y eliminarlo, dejando de ser monitorizado y evitando poder ser reactivado.

Tabla 86: Requisito del software RSF-59

Identificador: RSF-60	
Título:	Opción de crear alerta.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de crear una alerta para un equipo o plantilla a partir de uno o varios elementos monitorizados y una condición de alerta.

Tabla 87: Requisito del software RSF-60

Identificador: RSF-61	
Título:	Opción de eliminar alerta.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de eliminar una alerta definida en un equipo monitorizado o plantilla, dejando de notificar cuando se cumple su valor de alerta y evitando poder ser reactivado.

Tabla 88: Requisito del software RSF-61

Identificador: RSF-62	
Título:	Opción de desactivar alerta.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de desactivar una alerta en un equipo monitorizado o plantilla de tal forma que deje de notificar cuando se cumpla la

Identificador: RSF-62	
	condición de alerta.

Tabla 89: Requisito del software RSF-62

Identificador: RSF-63	
Título:	Opción de activar una alerta.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de activar una alerta en un equipo monitorizado o plantilla que haya sido previamente desactivada para que vuelva a notificar cuando se ha cumplido la condición de alerta.

Tabla 90: Requisito del software RSF-63

Identificador: RSF-64	
Título:	Opción de definir notificación.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de definir una notificación para un equipo monitorizado, plantilla o grupo a partir de una alerta de modo que cuando se cumpla su condición de alerta se envía un mensaje al administrador.

Tabla 91: Requisito del software RSF-64

Identificador: RSF-65	
Título:	Opción de notificar por email.

Identificador: RSF-65	
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de agregar una dirección de correo electrónico a una notificación donde será enviado un mensaje si se activa la alerta asociada.

Tabla 92: Requisito del software RSF-65

Identificador: RSF-66	
Título:	Opción de notificar por servicio de mensaje corto.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de agregar un número de teléfono a una notificación donde será enviado un mensaje de texto corto si se activa la alerta asociada.

Tabla 93: Requisito del software RSF-66

Identificador: RSF-67	
Título:	Opción de notificar por mensajería instantánea.
Tipo:	Funcional.
Prioridad:	Baja.
Necesidad:	Opcional.
Estabilidad:	Baja.
Descripción:	Se deberá proporcionar la opción de agregar una dirección de mensajería instantánea asociada a un servidor de XMPP a una notificación donde será enviado un mensaje si se activa la alerta asociada.

Tabla 94: Requisito del software RSF-67

Identificador: RSF-68	
Título:	Opción de desactivar notificación.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de desactivar una notificación para un equipo monitorizado, plantilla o grupo, de modo que si se activa la condición de alerta en alguno de ellos, no se enviará ningún mensaje.

Tabla 95: Requisito del software RSF-68

Identificador: RSF-69	
Título:	Opción de activar notificación.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de activar una notificación previamente desactivada para un equipo monitorizado, plantilla o grupo, de modo que si se activa la condición de alerta en alguno de ellos, se enviará el mensaje correspondiente.

Tabla 96: Requisito del software RSF-69

Identificador: RSF-70	
Título:	Opción de eliminar una notificación.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.

Identificador: RSF-70	
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de eliminar una alerta definida en el sistema, de modo que dejen de enviarse mensajes cuando se cumpla la condición de alerta asociada a la notificación y evitando poder ser reactivada.

Tabla 97: Requisito del software RSF-70

Identificador: RSF-71	
Título:	Pantalla de alertas.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en la que se muestren todas las alertas definidas para un determinado equipo monitorizado o plantilla, que permitirá la gestión de las mismas.

Tabla 98: Requisito del software RSF-71

Identificador: RSF-72	
Título:	Pantalla de notificaciones.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en la que se muestren todas las notificaciones definidas en el sistema permitiendo la gestión de las mismas.

Tabla 99: Requisito del software RSF-72

Identificador: RSF-73	
Título:	Opción de crear plantilla.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de crear plantillas de la misma manera que equipos monitorizados. Las plantillas contendrán un conjunto de elementos monitorizados, alertas y gráficas.

Tabla 100: Requisito del software RSF-73

Identificador: RSF-74	
Título:	Opción de eliminar plantilla.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de eliminar una plantilla del sistema de monitorización dejando de ser aplicada en todos los equipos monitorizados que la utilizan.

Tabla 101: Requisito del software RSF-74

Identificador: RSF-75	
Título:	Opción de aplicar plantilla.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Baja.
Descripción:	Se deberá proporcionar la opción de aplicar una plantilla creada a un equipo monitorizado a todos los

Identificador: RSF-75	
	equipos monitorizados de un grupo, adquiriendo esos equipos todos los elementos monitorizados, alertas y gráficas definidos en la plantilla.

Tabla 102: Requisito del software RSF-75

Identificador: RSF-76	
Título:	Opción de quitar plantilla.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Baja.
Descripción:	Se deberá proporcionar la opción de suprimir una plantilla de un equipo monitorizado, de manera que todos los elementos monitorizados, alertas y gráficas modificados en la plantilla dejen de ser aplicables en el equipo.

Tabla 103: Requisito del software RSF-76

Identificador: RSF-77	
Título:	Opción de crear gráfica.
Tipo:	Funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de crear una gráfica para un equipo monitorizado o plantilla a partir de uno o varios elementos monitorizados.

Tabla 104: Requisito del software RSF-77

Identificador: RSF-78	
Título:	Opción de eliminar gráfica.

Identificador: RSF-78	
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar la opción de eliminar una gráfica de un equipo monitorizado o plantilla.

Tabla 105: Requisito del software RSF-78

Identificador: RSF-79	
Título:	Pantalla de gráficas.
Tipo:	Funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deberá proporcionar una pantalla en al que se muestren todas las gráficas definidas para un equipo monitorizado o plantilla, permitiendo la gestión de las mismas.

Tabla 106: Requisito del software RSF-79

4.2.2.2. Requisitos no funcionales

Identificador: RSNF-01	
Título:	Máquinas virtuales disponibles desde cualquier ubicación mediante escritorio remoto.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Las máquinas virtuales deberán estar disponibles

[redacted]	para los teletrabajadores desde cualquier equipo con conexión a Internet, mediante escritorio remoto RDP.

Tabla 107: Requisito del software RSNF-01

Identificador: RSNF-02	
Título:	Máquinas virtuales disponibles en cualquier momento.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Las máquinas virtuales deberán estar disponibles para los teletrabajadores en cualquier momento.

Tabla 108: Requisito del software RSNF-02

Identificador: RSNF-03	
Título:	Acceso a máquinas virtuales restringido.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El acceso a las máquinas virtuales estará restringido a direcciones IP de la UC3M, siendo imprescindible que esté disponible un servicio corporativo de VPN para su acceso.

Tabla 109: Requisito del software RSNF-03

Identificador: RSNF-04	
Título:	Máquinas virtuales monitorizadas.
Tipo:	No funcional.
Prioridad:	Media.

Identificador: RSNF-04	
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Las máquinas virtuales deberán estar monitorizadas en todo momento desde un sistema externo.

Tabla 110: Requisito del software RSNF-04

Identificador: RSNF-05	
Título:	Máquinas virtuales en dominio UC3M.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Las máquinas virtuales deberán estar incluidas en el dominio de la UC3M.

Tabla 111: Requisito del software RSNF-05

Identificador: RSNF-06	
Título:	Máquinas virtuales en DNS.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Las máquinas virtuales deben tener asociado un nombre DNS para poder emplearlo en lugar de la dirección IP.

Tabla 112: Requisito del software RSNF-06

Identificador: RSNF-07	
Título:	Máquinas virtuales con Windows XP.
Tipo:	No funcional.

Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El sistema operativo de las máquinas virtuales deberá ser Windows XP.

Tabla 113: Requisito del software RSNF-07

Identificador: RSNF-08	
Título:	Sistema de virtualización de software libre.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El sistema de virtualización debe ser software libre.

Tabla 114: Requisito del software RSNF-08

Identificador: RSNF-09	
Título:	Tipo de virtualización.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El tipo de virtualización empleada debe ser completa para proporcionar el mayor rendimiento posible.

Tabla 115: Requisito del software RSNF-09

Identificador: RSNF-10	
Título:	Múltiples servidores.
Tipo:	No funcional.
Prioridad:	Alta.

Identificador: RSNF-10	
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Para proporcionar la posibilidad de migración de máquinas virtuales y asegurar alta disponibilidad se requieren al menos dos servidores en clúster.

Tabla 116: Requisito del software RSNF-10

Identificador: RSNF-11	
Título:	Almacenamiento externo en red.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Para proporcionar la posibilidad de migración de máquinas virtuales y asegurar alta disponibilidad se requiere almacenamiento en red.

Tabla 117: Requisito del software RSNF-11

Identificador: RSNF-12	
Título:	Acceso a servidores restringido.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El acceso a los servidores estará restringido a direcciones IP de la UC3M.

Tabla 118: Requisito del software RSNF-12

Identificador: RSNF-13	
Título:	Servidores monitorizados.

Identificador: RSNF-13	
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Los servidores deberán estar monitorizados en todo momento desde un sistema externo.

Tabla 119: Requisito del software RSNF-13

Identificador: RSNF-14	
Título:	Servidores en DNS.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Los servidores deben tener asociado un nombre DNS para poder emplearlo en lugar de la dirección IP.

Tabla 120: Requisito del software RSNF-14

Identificador: RSNF-15	
Título:	Agente para Windows
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Debe haber una aplicación que se ejecute en el equipo monitorizado con sistema operativo Windows para medir elementos y enviar los datos a un servidor indicado.

Tabla 121: Requisito del software RSNF-15

Identificador: RSNF-16	
Título:	Agente para Unix.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Debe haber una aplicación que se ejecute en el equipo monitorizado con sistema operativo compatible con Unix para medir elementos y enviar los datos a un servidor indicado.

Tabla 122: Requisito del software RSNF-16

Identificador: RSNF-17	
Título:	Máquinas con agente instalado.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Todo equipo monitorizado debe tener un agente instalado.

Tabla 123: Requisito del software RSNF-17

Identificador: RSNF-18	
Título:	Backup no intrusivo.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	La realización de las funciones de backup debe realizarse afectando lo menos posible a la actividad del teletrabajador. Realizándose por las noches.

Tabla 124: Requisito del software RSNF-18

Identificador: RSNF-19	
Título:	Puntos de restauración.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deben de realizar los backups necesarios para ofrecer al usuario al menos dos puntos de restauración de la máquina virtual a la semana.

Tabla 125: Requisito del software RSNF-19

Identificador: RSNF-20	
Título:	Redundancia de máquinas virtuales.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se deben replicar las máquinas virtuales para que ante la caída en uno de los nodos del servidor de Teletrabajo no se interrumpa el servicio dado por las máquinas virtuales hospedadas en ese nodo.

Tabla 126: Requisito del software RSNF-20

Identificador: RSNF-21	
Título:	Recuperación de máquina.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Baja.
Descripción:	Las caídas en las máquinas virtuales deben ser detectadas y recuperadas en menos de un 1 día

Identificador: RSNF-21	
	laborable.

Tabla 127: Requisito del software RSNF-21

Identificador: RSNF-22	
Título:	Backup corporativo actualizado.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	El backup corporativo debe salvar el estado más reciente de las máquinas virtuales, al menos salvará las máquinas con el estado que tenían al final de la jornada del día que se realice.

Tabla 128: Requisito del software RSNF-22

Identificador: RSNF-23	
Título:	Redundancia de backup.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Los backups de las máquinas virtuales se almacenarán en al menos dos servidores diferentes para poder recuperar el backup en caso de que falle uno de los dos.

Tabla 129: Requisito del software RSNF-23

Identificador: RSNF-24	
Título:	Registro de intentos de acceso.
Tipo:	No funcional.
Prioridad:	Media.

Identificador: RSNF-24	
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Se mantendrá un fichero de texto en el que se registran todos los accesos a través de la red que ha sufrido el servidor de Teletrabajo.

Tabla 130: Requisito del software RSNF-24

Identificador: RSNF-25	
Título:	Comprobar integridad de elementos del servicio.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Conveniente.
Estabilidad:	Media.
Descripción:	Se realizarán funciones resumen de ficheros críticos, como núcleo del sistema o claves almacenadas para asegurar que no han sido modificados.

Tabla 131: Requisito del software RSNF-25

Identificador: RSNF-26	
Título:	Privacidad de las sesiones de teletrabajo.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	La sesión de escritorio remoto de teletrabajador es segura y no puede ser visualizada por terceras personas.

Tabla 132: Requisito del software RSNF-26

Identificador: RSNF-27	
Título:	Identificación de servidores.

Identificador: RSNF-27	
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Los servidores de Teletrabajo contarán con certificados electrónicos que aseguran que no han sido suplantados.

Tabla 133: Requisito del software RSNF-27

Identificador: RSNF-28	
Título:	Extensión de plantillas de monitorización.
Tipo:	No funcional.
Prioridad:	Media.
Necesidad:	Conveniente.
Estabilidad:	Alta.
Descripción:	Se pueden definir jerarquías en las plantillas mediante operaciones de herencia y generalización. De modo que al definir una plantilla como heredera de otra, implica que la plantilla definida adquirirá todas las características definidas en la original. Además cualquier modificación realizada en la plantilla original, se aplicará automáticamente en las plantillas herederas.

Tabla 134: Requisito del software RSNF-28

Identificador: RSNF-29	
Título:	Coherencia entre plantillas y equipos.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Si se aplica una plantilla a un equipo monitorizado,

Identificador: RSNF-29	
	cualquier cambio realizado en la plantilla debe aplicarse automáticamente en el equipo.

Tabla 135: Requisito del software RSNF-29

Identificador: RSNF-30	
Título:	Código de colores.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	Las alertas contarán con un icono gráfico que cambia de color según el estado de la alerta, empezando en verde si funciona correctamente, pasando por amarillo si es una advertencia o rojo si existe un problema.

Tabla 136: Requisito del software RSNF-30

Identificador: RSNF-31	
Título:	Acceso de monitorización.
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El servidor de monitorización debe tener acceso por red a todos los equipos monitorizados y todos los equipos monitorizados deben tener acceso al servidor de monitorización.

Tabla 137: Requisito del software RSNF-31

Identificador: RSNF-32	
Título:	Servicio de monitorización independiente.

Identificador: RSNF-32	
Tipo:	No funcional.
Prioridad:	Alta.
Necesidad:	Imprescindible.
Estabilidad:	Alta.
Descripción:	El servidor de monitorización debe estar localizado en un equipo externo e independiente de los servidores de Teletrabajo.

Tabla 138: Requisito del software RSNF-32

4.3. Diseño arquitectónico

En este apartado se presenta el diseño arquitectónico del sistema. Primeramente se identificarán los nodos de computación mediante un diagrama de despliegue. A continuación se definirá una plataforma hardware concreta, en base a los nodos identificados. Finalmente se identificarán los componentes software que serán desplegados sobre la plataforma propuesta.

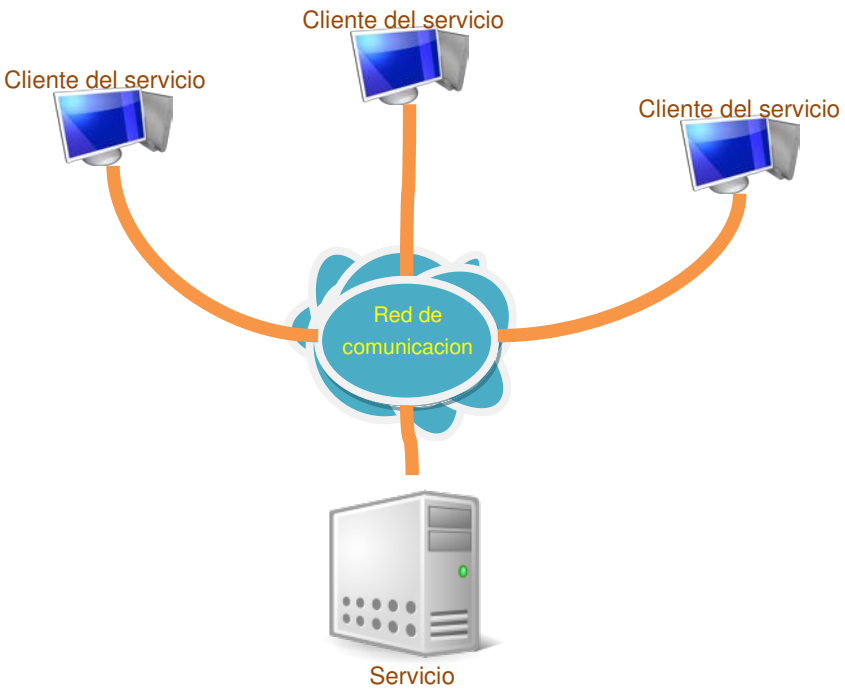

Ilustración 47 Esquema de la arquitectura cliente-servidor

En la *Ilustración 47 Esquema de la arquitectura cliente-servidor* se muestra la arquitectura que va a seguir el sistema. La arquitectura cliente-servidor cuenta con un nodo central, denominado servidor al que se conectan distintos nodos denominados clientes, estando el sistema dividido. Los nodos pueden estar situados en distintos o en el mismo equipo y la carga computacional puede distribuirse entre el cliente y el servidor dependiendo del problema a resolver.

En el caso que ocupa ahora, la arquitectura cliente-servidor está presente en diversos aspectos del sistema: los teletrabajadores se conectan al servidor de teletrabajo mediante un cliente que les permite ver la máquina virtual, o los agentes de monitorización (clientes) recopilan datos y se los envían al servidor de monitorización que los procesa.

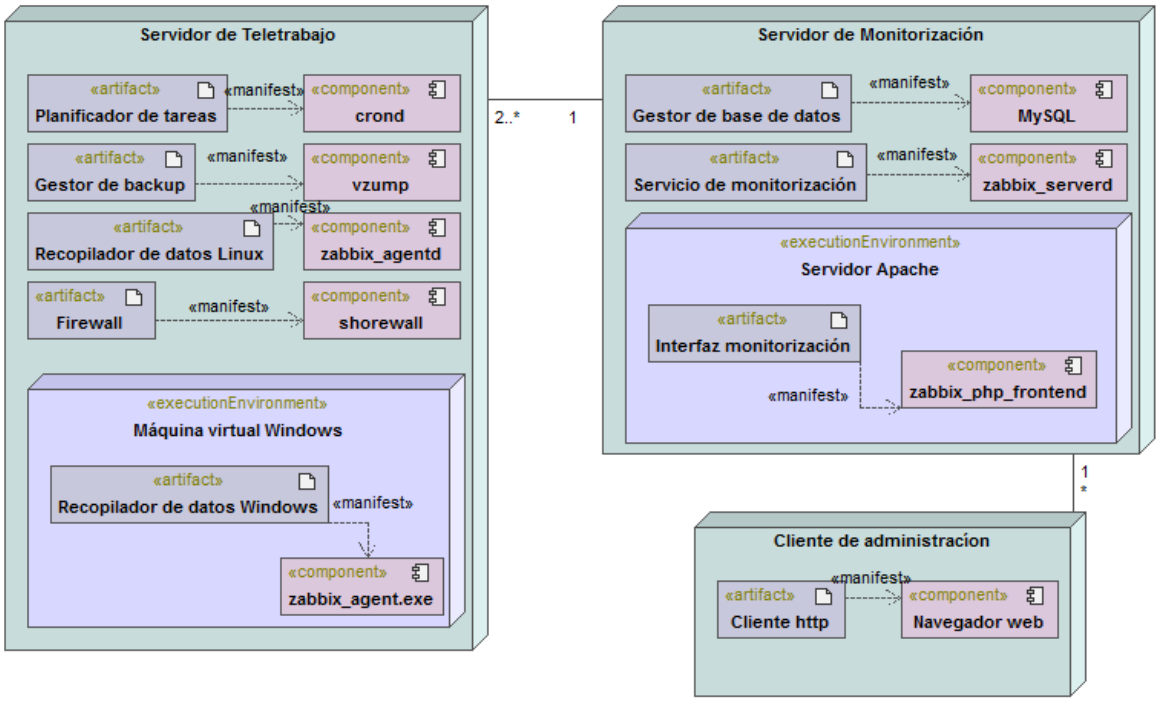

Ilustración 48 Arquitectura del sistema

En la *Ilustración 48 Arquitectura del sistema* se puede observar el diagrama de despliegue de las partes de backup, seguridad y monitorización que abarca este proyecto para el Servicio de Teletrabajo de la Universidad Carlos III. Únicamente se muestran las funciones mencionadas dado que la funcionalidad de virtualización queda desarrollada en el proyecto anterior (Gil Bázquez, 2011).

El diagrama únicamente presenta cinco de los seis nodos implicados en el sistema, siendo el sexto el equipo que utiliza el teletrabajador para acceder a su máquina virtual.

Comenzando por el nodo "Cliente de administración", con este nombre se representa el equipo de escritorio que utiliza el administrador de monitorización para acceder al

servidor de monitorización. Dado que el servidor de monitorización cuenta con una interfaz gráfica accesible remotamente mediante el protocolo Http, únicamente es necesario que el "Cliente de administración" cuente con un navegador web de su preferencia.

El nodo "Servidor de monitorización" representa la máquina virtual (externa al servidor de Teletrabajo) dedicada únicamente a recuperar información de la red monitorizada, almacenar esa información y mostrársela al administrador de monitorización. Como se puede observar, es un nodo separado del "Servidor de Teletrabajo" al no formar parte del mismo sistema, lo que es imprescindible para asegurar que el servicio de monitorización es capaz de cumplir con su objetivo de alertar al administrador sobre incidencias en el Servicio de Teletrabajo. Si el "Servidor de monitorización" estuviese incluido dentro de la estructura del "Servidor de Teletrabajo", cualquier fallo que ocurriese en este servidor podría afectar al de monitorización, como pueden ser caídas del sistema, caídas de la red, caída en los procesos de virtualización etc. Cualquiera de estos fallos comentados inhabilitaría el servicio de monitorización, el cual no podría notificar al administrador sobre los fallos en el Servicio de Teletrabajo, corriendo el riesgo de que no se detecten y reduciendo el tiempo de respuesta hasta poder restaurar el sistema.

El nodo de "Servidor de monitorización" cuanta con tres componentes cuya funcionalidad es análoga a la de los integrantes de la arquitectura modelo-vista-controlador. A continuación se detallan los componentes que forman parte de este nodo:

- **Zabbix_php_frontend**: es un programa escrito en php que se encarga de recuperar los datos de la base de datos y mostrarlos mediante una interfaz gráfica al administrador de monitorización. También se encarga de la representación gráfica de esos datos mediante gráficas y de mostrar notificaciones visuales cada vez que se cumple la condición definida en una alerta.
- **Apache server**: es el entorno de ejecución encargado de atender las peticiones que le envía el administrador de monitorización mediante el protocolo http, respondiendo a sus peticiones enviando la página html de la interfaz de Zabbix solicitada. El administrador de monitorización solicita páginas web html para poder visualizarlas en su navegador web, sin embargo el código de la interfaz de Zabbix está escrito en el lenguaje php. El entorno de ejecución cuenta con el intérprete de php que se encarga de ejecutar el código de "zabbix_php_frontend" para obtener como resultado una página en formato html compatible con el navegador del administrador de monitorización. Tener la interfaz escrita en php (en lugar de html que evitaría

este paso de transformación) permite tener una interfaz con contenido dinámico que cambia en función de las alertas que surjan en el sistema y nuevos valores que se recopilan de los agentes, siendo de esta manera una interfaz web estática escrita en html inútil para nuestro propósito.

- **MySQL**: la base de datos MySQL es la pieza central del sistema de monitorización al reducirse este a un sistema que recopila y almacena datos para mostrarlos más adelante. No es imprescindible que la base de datos sea MySQL, valiéndonos cualquier otro software como PostgreSQL, InnoDB, MyISAM, RAID10 o Fast. Cada base de datos se adapta mejor a una red monitorizada de distinto tamaño dependiendo del volumen de datos que tenga que importar continuamente, sin embargo MySQL aparece en la documentación como una base de datos tanto para redes muy pequeñas (de tan solo 10 equipos), hasta redes grandes (de más de 1000 equipos), siendo una solución ideal si se desea que el sistema de monitorización sea escalable (como es la intención en un futuro que no se limite a monitorizar los equipos del Servicio de Teletrabajo). Otra ventaja de escoger MySQL como base de datos es su gran popularidad siendo fácil encontrar futuros administradores que sean capaz de utilizarla.
- **Zabbix_server**: este proceso será el encargado de estar continuamente a la escucha de las conexiones que realizarán con el servidor de monitorización todos los procesos agente, distribuidos en los equipos monitorizados. No se limita a recibir conexiones pasivamente, sino que, para algunos elementos monitorizados, es zabbix_server el encargado de preguntar activamente por esos valores a los agentes. Una vez ha obtenido todos los datos que debe recopilar, es el encargado de hacer los cálculos pertinentes e insertar los datos registrados y los calculados en la base de datos.

En el nodo “Servidor de Teletrabajo”, a diferencia del “Servidor de monitorización”, se mezclan componentes que satisfacen distintas funcionalidades. Así, aparecen componentes que sirven para virtualizar equipos (no representados), componentes encargados del control de acceso al servidor, otros encargados de realizar copias de seguridad de las máquinas virtualizadas y, por último, un componente encargado de la extracción de datos del servicio para la monitorización.

- **Crond**: es un demonio del sistema Linux que se dedica a ejecutar tareas en unos intervalos de tiempo y frecuencia definidos en un fichero de configuración, permitiendo programar la ejecución automática de los programas deseados los días y horas que sean necesarios.

- **Vzdump**: es el componente encargado de realizar backups de las máquinas virtuales en ejecución del Servicio de Teletrabajo, ocultando al usuario el proceso al recibir únicamente un identificador de máquina virtual para encargarse de, no solo devolver la imagen del disco duro y el fichero de configuración de esa máquina virtual, sino también del mantenimiento de un número máximo de backups, eliminando los más obsoletos.
- **Shorewall**: el componente Shorewall únicamente se ejecuta durante el arranque del sistema (o cuando el usuario lo lanza explícitamente), no siendo un proceso presente durante el funcionamiento del servidor. Cuando se ejecuta este componente se encarga de leer unos ficheros de configuración que definen el control de acceso al sistema y pasa a configurar automáticamente el firewall del núcleo de Linux.
- **Zabbix_agentd**: este componente representa al agente de monitorización del sistema, encargado de recopilar datos sobre el sistema en el que se ejecuta, contactar con el servidor de monitorización y enviarle los datos. También se encarga de recopilar datos bajo petición del servidor de monitorización para enviárselos. Dispone de diversos métodos para recopilar información como ejecutar comandos y almacenar la salida del comando, o leer un fichero de log, pero ciertos comandos y logs sólo pueden ser accedidos por el usuario "root" del sistema, siendo necesario ofrecer estos mismos permisos de acceso al agente.

Dentro del nodo "Servidor de Teletrabajo" se encuentran varias instancias de las máquinas virtuales utilizadas por los teletrabajadores. Esas máquinas son virtualizadas mediante KVM que utiliza virtualización completa, siendo por tanto equipos totalmente independientes e inaccesibles por el agente de monitorización de la máquina anfitriona. Las máquinas virtuales utilizadas para proveer el Servicio de Teletrabajo cuentan con sistema operativo Microsoft Windows XP, siendo necesario por tanto el uso de un agente distinto al del servidor que haya sido compilado para funcionar en Windows.

4.3.1. Infraestructura Hardware

Como se ha visto en el apartado *4.3 Diseño arquitectónico*, hay dos nodos referentes a los servidores que representan sistemas distintos, cada uno de los cuales dispondrá de una plataforma distinta sobre la que se ejecutará. En concreto, el servidor de Teletrabajo es un clúster configurado a partir de la red local que cuenta con 2 equipos iguales de las siguientes características:

- Servidor: HP ProLiant BL460c.
- Procesador: Intel Xeon X5650 2,67GHz con 6 núcleos.
- Memoria RAM: 32 GB.
- Disco duro: HP 500 GB.
- Tarjeta de red: 2 interfaces de red de 10 Gb.

Además, los servidores de Teletrabajo disponen de un espacio de almacenamiento en red de 500 GB, proporcionado por una cabina EMC CLARiiON y accesible a través del protocolo iSCSI. Este almacenamiento es donde se localizarán los discos duros de las máquinas virtuales.

El servidor en el que está localizado el servicio de monitorización es una única máquina con las siguientes características:

- Servidor: HP Proliant DL380 G5.
- Procesados: Intel Xeon E5440 2,83 GHz con 2 núcleos.
- Memoria RAM: 8GB.
- Disco duro: 150 GB.
- Tarjeta de red: 2 interfaces Broadcom NetXtreme II Gigabit Ethernet.

No obstante, el equipo no está totalmente dedicado al servicio de monitorización ya que alberga distintas máquinas virtuales, entre ellas el servidor de monitorización. Los recursos dedicados a la máquina virtual de monitorización son:

- Procesador: 1 socket con 2 núcleos.
- Memoria RAM: 2 GB.
- Disco duro: 25 GB.
- Tarjeta de red: rtl8139.

4.3.2. Infraestructura Software

A continuación, se pasa a detallar el funcionamiento del conjunto de componentes anteriormente descritos, los cuales parecían inconexos debido a que se trataba de la vista de despliegue. En este apartado se definirá cómo se relacionan los componentes entre ellos para conseguir que el sistema consiga cumplir con su cometido.

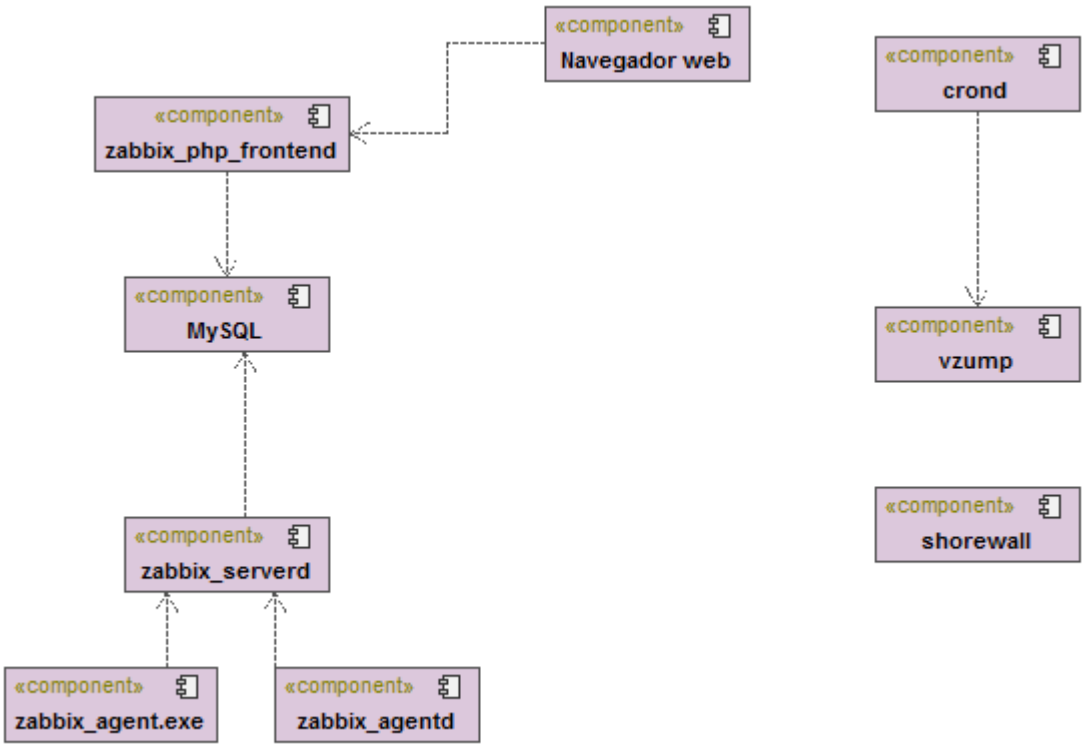

Ilustración 49 Diagrama de componentes

En *Ilustración 49 Diagrama de componentes* se observan las dependencias entre los distintos componentes que forman parte de la arquitectura detallada en el punto *4.3 Diseño arquitectónico*.

Hay tres partes inconexas, en cuanto a dependencias con otros componentes, que representan las tres funcionalidades en las que se separa la seguridad del Servicio de Teletrabajo. Sólo se puede hablar de que se está ofreciendo un servicio de seguridad si están funcionando cada una de las tres partes (control de acceso, backup y monitorización), pero si alguna de las partes deja de funcionar, el resto de las partes continuarán funcionando sin ningún problema.

El primer bloque comprende a la funcionalidad de control de acceso, que únicamente abarca el componente Shorewall. Este componente despliega un script de configuración que realiza llamadas al framework que ofrece *Netfilter/iptables*. Encargándose a partir de ese momento el núcleo de Linux del control de acceso al sistema siendo transparente para el resto del sistema, al no haber ningún proceso en ejecución por el que deba de pasar previamente el tráfico de red para filtrarlo.

La elección de utilizar *Netfilter* como el firewall del sistema es una decisión sencilla, ya que si no disponemos de un firewall hardware, encargado de filtrar el tráfico antes de entrar al servidor, se debe hacer mediante software. Además se manejarán dispositivos conectados dentro de una subred virtual, localizada dentro del

propio servidor de Teletrabajo, haciendo imposible conectar un firewall físico en esa red. Respecto a las alternativas de firewalls en Linux, el de uso más extendido y con más documentación que se puede encontrar es *Netfilter*, ya que se provee dentro del núcleo del sistema, siendo necesario únicamente configurarlo. El resto de alternativas de firewalls para Linux, además de requerir un proceso adicional que ejecutar (y asegurarse de que siempre está en ejecución), normalmente están orientados a usuarios de equipos de escritorio, ofreciendo una interfaz gráfica que permite configurarlos fácilmente, pero los servidores de Teletrabajo no cuentan con un entorno de trabajo gráfico, no siendo posible el uso de estas interfaces.

Por su parte, *Iptables* no es sencillo de configurar y es especialmente difícil de mantener, siendo necesario que el administrador conozca las reglas que hay definidas y cuales puede añadir o eliminar. La elección de *Shorewall* se realiza para simplificar esa tarea de configuración y mantenimiento pudiendo, gracias a él, definir en un fichero de texto debidamente comentado cuál es la funcionalidad que se espera del firewall.

El segundo bloque comprende dos componentes: crond y vzdump. Crond es un demonio del sistema que ejecuta la tarea definida en unos intervalos de tiempo también definidos, haciéndolo idóneo para programar la realización del backup unos determinados días a unas determinadas horas. La tarea que tendrá que ejecutar es la invocación de una serie de scripts (que deben ser desarrollados) que, mediante ejecuciones del programa *vzdump*, consigan realizar y almacenar copias de las máquinas virtuales hospedadas en el servidor.

Para terminar, el último bloque es el responsable de la función de monitorización. La monitorización comienza por la medición de elementos monitorizados por los agentes instalados en diversos sistemas operativos: zabbix_agent.exe y zabbix_agentd en sistemas compatibles con Unix. Una vez recopilada la información indicada, los agentes se conectan con el componente zabbix_serverd y le transmiten esa información. El componente zabbix_serverd transforma la información recogida en sentencias para su inserción en la base de datos del sistema de monitorización, que en este caso está gestionada por el componente MySQL.

Por otro lado, el administrador del sistema de monitorización, para poder acceder al sistema, utiliza el navegador web instalado en su equipo de escritorio, que se conecta al servidor de monitorización mediante el protocolo http. La conexión http es atendida por el entorno de ejecución del servidor Apache que, para poder ofrecer una respuesta, invoca el código de zabbix_php_frontend. El componente zabbix_php_frontend se encarga de extraer todos los datos de monitorización de la base de datos MySQL implicados en la consulta del administrador, organizándolos y

presentándolos en un documento html que será entregado al navegador web como respuesta a su consulta.

4.4. Diseño detallado

Una vez definida la arquitectura y componentes que formarán parte del sistema, se pasará a diseñar cómo se realizarán cada una de las funciones implicadas en la seguridad del Servicio de Teletrabajo.

En primer lugar se analizarán las conexiones de red del Servicio de Teletrabajo y las zonas en las que se divide esa red, para identificar aquellas que suponen un mayor riesgo y configurar así el firewall en consecuencia.

Más adelante se propondrán varios diseños sobre la funcionalidad de backup, que habrá que probar para compararlos en términos de eficiencia y adecuación a los requisitos software, eligiéndose la mejor opción a implementar.

El último paso será el diseño de la funcionalidad de monitorización, donde se determinará que parámetros de los equipos son necesarios controlar, en cuáles hay que fijar límites de alarma y cuándo será necesario el envío de alertas al administrador.

4.4.1. Diseño del control de acceso

Antes de poder diseñar la configuración del firewall hay que analizar la configuración de red del Servicio de teletrabajo, es decir, las interfaces que tiene y cómo y dónde están conectadas.

Los servidores de teletrabajo cuentan con dos interfaces de red: eth0, conectada a la red de la Universidad, a través de la cual se realiza la conexión a internet, y por otro lado la interfaz eth1, a través de la cual se realiza la conexión con el disco duro en red iSCSI.

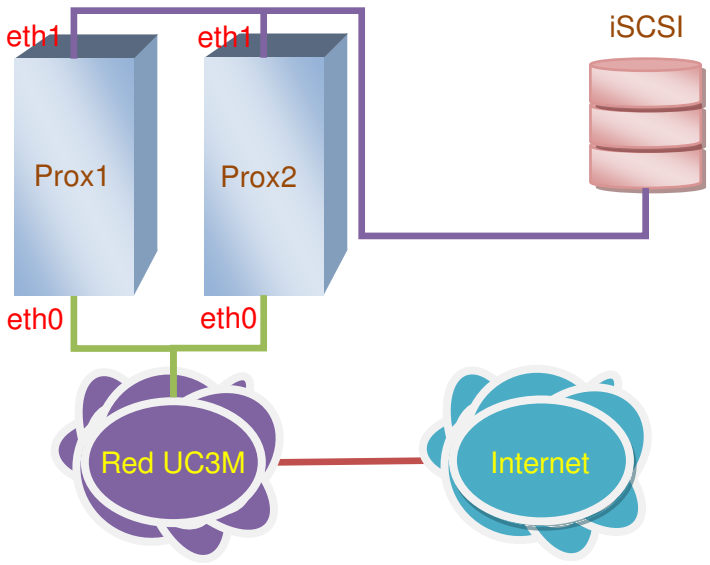

Ilustración 50 Esquema de la red del servidor de Teletrabajo

El esquema de la *Ilustración 50 Esquema de la red del servidor de Teletrabajo* muestra a dónde están conectados físicamente los servidores de Teletrabajo. Sin embargo, hay que tener en cuenta que hay máquinas virtuales dentro de los servidores de Teletrabajo que están también conectadas a la red, siendo necesario definir interfaces virtuales y un bridge para poder conectarlas. A partir de ahora olvidaremos que tenemos varios servidores de Teletrabajo para tener la visión de un único servidor, dado que ambos servidores tienen configuraciones idénticas.

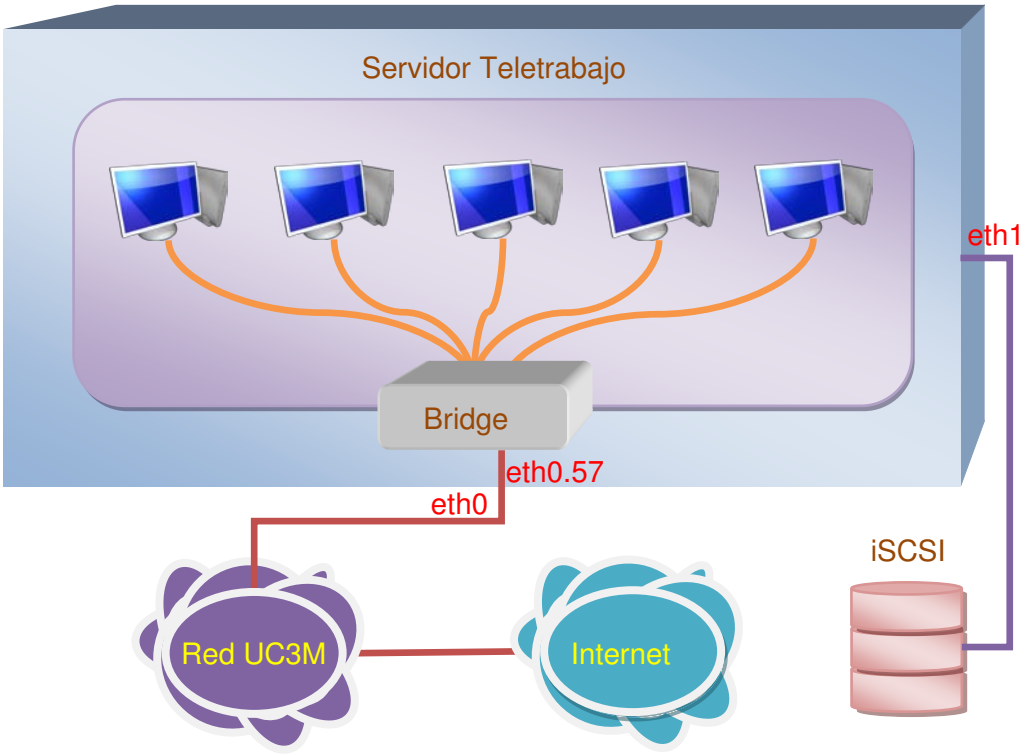

Ilustración 51 Esquema de la red del servidor de Teletrabajo 2

El primer paso para configurar el firewall es dividir la red en zonas que sirvan para simplificar la definición del control de acceso a los equipos, de esta manera se comienza identificando tres grandes zonas: el servidor de Teletrabajo ("FW") con su disco de almacenamiento ("ISCSI") y el resto del mundo ("World"). Las máquinas virtuales hospedadas en el servidor de Teletrabajo, a pesar de estar ejecutando dentro de él, son equipos independientes conectados a una red común, y por tanto no se debe presuponer que su actividad tiene que ser inocua para el servidor, por lo que todas esas máquinas, y la vlan a la que pertenecen, serán considerados partes de la zona externa "World".

La zona "World" es demasiado genérica para ser tratada como una única, siendo conveniente subdividirla en otras tres: la red de la Universidad Carlos III, es decir todas las direcciones con dirección IP de la red 162.117.0.0/16 que se etiquetará como "UC3M". Las máquinas virtuales contenidas en el servidor de Teletrabajo se separarán en una zona distinta denominada "DMZ" que se corresponde con las máquinas de la subred 163.117.180.0/24, subred a la que se conectan las máquinas virtuales. Finalmente, la última zona es la denominada "net" que contiene el resto de la red, es decir lo que se pude considerar internet o la red externa a la UC3M.

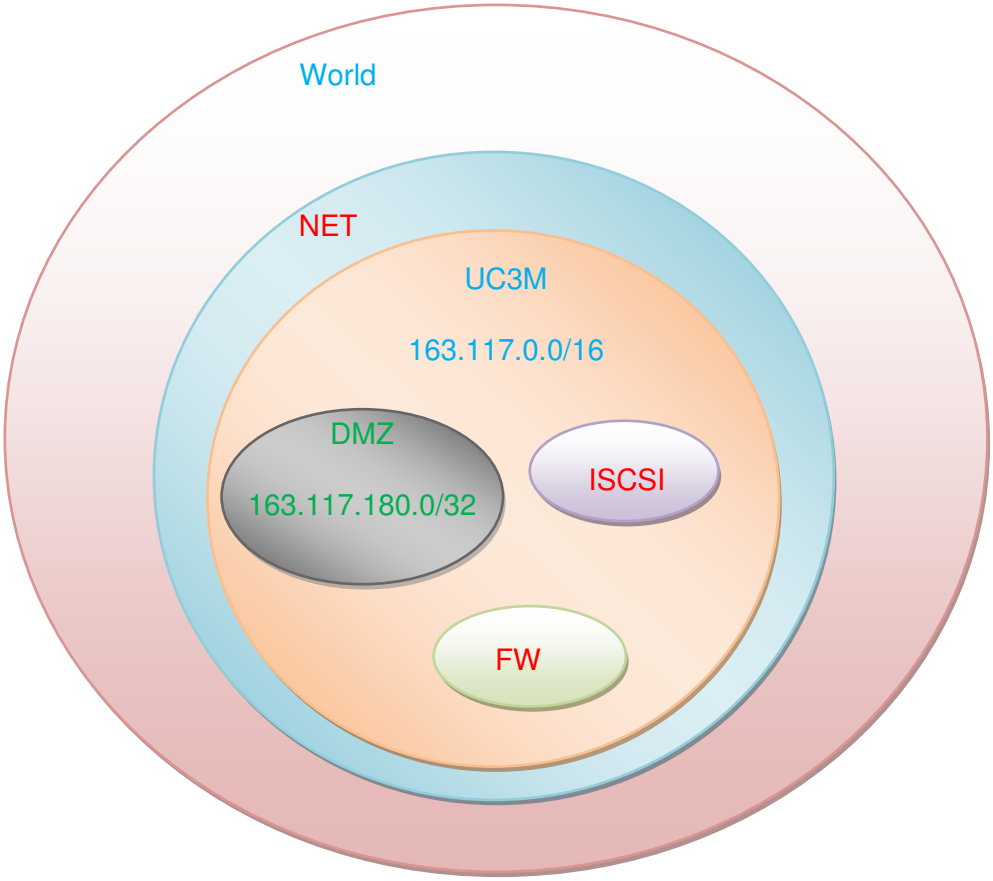

Ilustración 52 División en zonas de la red de Teletrabajo

La configuración del firewall se realiza discriminando cada conjunto de paquetes por su dirección de origen, dirección de destino, puerto de origen o puerto de destino, y definiendo cuál de las tres acciones se aplicará cuando se reciba un paquete. Las acciones posibles son: **aceptarlo**, es decir, dejarlo pasar para que llegue a su destino; **rechazarlo**, que implica cortar el paso del paquete y mandar un mensaje al remitente informándole que ha sido rechazado; y finalmente **tirarlo**, que igual que el anterior corta el paso del paquete descartándolo, pero sin informar que ha sido descartado, por lo que el remitente cree que se ha perdido.

En primer lugar definimos como política por defecto del firewall rechazar todos los paquetes. Así, si no se encuentra ninguna política aplicable, siempre se tiene la certeza de que no se permitirá ninguna conexión, reduciendo de este modo los riesgos de intrusión en el sistema. Sería más seguro utilizar una política que tirase todos los paquetes en lugar de rechazarlos, para así no dar indicios a un posible atacante de que en esa dirección haya un servidor. Pero, puesto que la mayoría de zonas pertenecen a la Universidad, y desde la Universidad se accederá a los servicios de Teletrabajo, se

opta por rechazar paquetes. De este modo, el teletrabajador que se intente conectar de forma errónea, p.e. a un puerto distinto, recibirá una notificación de que el servicio al que está intentando acceder es el equivocado.

Tras definir la política por defecto, hay que asegurar que el servidor puede acceder a su unidad de almacenamiento. El servidor de Teletrabajo debe poder acceder sin restricciones al disco iSCSI en red, por lo que todo el tráfico entre las zonas FW e ISCSI debe ser aceptado, sin rechazar o tirar ningún paquete. Además, todo el tráfico saliente desde la zona FW del servidor hacia el mundo (World) ha de estar aceptado, permitiendo así conexiones http o ftp con servidores en internet para realizar descargas de actualizaciones o descargas de software mediante el gestor de paquetes. En cuanto a las conexiones provenientes de la zona NET, los paquetes recibidos son inmediatamente tirados, dado que es la zona más peligrosa y desde donde puede ser explotada cualquier vulnerabilidad del servidor, ya que es donde potencialmente se localizarán los atacantes. Al aplicar como política tirar los paquetes que vienen de la zona NET, se impone sobre la política por defecto de rechazarlos, no emitiendo ninguna respuesta cuando se realiza un intento de conexión desde el exterior, y por tanto "ocultando" el servidor.

Desde la red interna de la Universidad se realizan distintos tipos de acceso al servidor de Teletrabajo: conexiones RDP para acceder a las máquinas virtuales, sesiones ssh para administrar el servidor, conexiones entrantes y salientes para las actividades del servicio de monitorización, conexiones para la realización del backup corporativo, etc. Son tantas y variadas las conexiones que recibe desde la zona UC3M que se acepta el tráfico proveniente de ella hacia la zona FW por simplicidad durante la fase de desarrollo y pruebas, para más adelante, cuando esté suficientemente establecido el sistema y delimitados los servicios y conexiones requeridos cambiar la política a rechazar todo el tráfico, excepto a los puertos de los servicios que se van a ofrecer.

Por último sólo queda definir cuáles son las conexiones permitidas para las máquinas virtuales incluidas dentro de la zona DMZ. En primer lugar, es necesario poder permitir a los teletrabajadores conectarse a sus máquinas virtuales a través de una conexión RPD, según el requisito software definido en *Tabla 109: Requisito del software RSNF-03* únicamente se permitirá el acceso de los teletrabajadores a través de la red de la Universidad, ya sea porque el equipo cliente está localizado dentro de la Universidad o utiliza una conexión VPN para obtener una dirección IP de la red de la Universidad. Por tanto únicamente es necesario configurar el firewall para que acepte el tráfico proveniente de la zona UC3M hacia la zona DMZ. Puesto que no sabemos qué usos o servicios utilizarán los teletrabajadores desde sus máquinas virtuales, no se puede restringir el acceso únicamente al puerto de RDP. De todas

formas, las máquinas virtuales del Servicio de Teletrabajo cuentan con un firewall software instalado que permite personalizar al usuario si desea permitir el paso a algunas conexiones. Finalmente, para permitir a los teletrabajadores acceder a internet desde sus máquinas virtuales, se acepta todo el tráfico saliente de la zona DMZ a la zona NET, lo cual no sólo se limita a acceso a internet, sino también a la red de la Universidad, al ser la zona UC3M un subconjunto de la zona NET como se puede observar en *Ilustración 53 Esquema de política del firewall.*

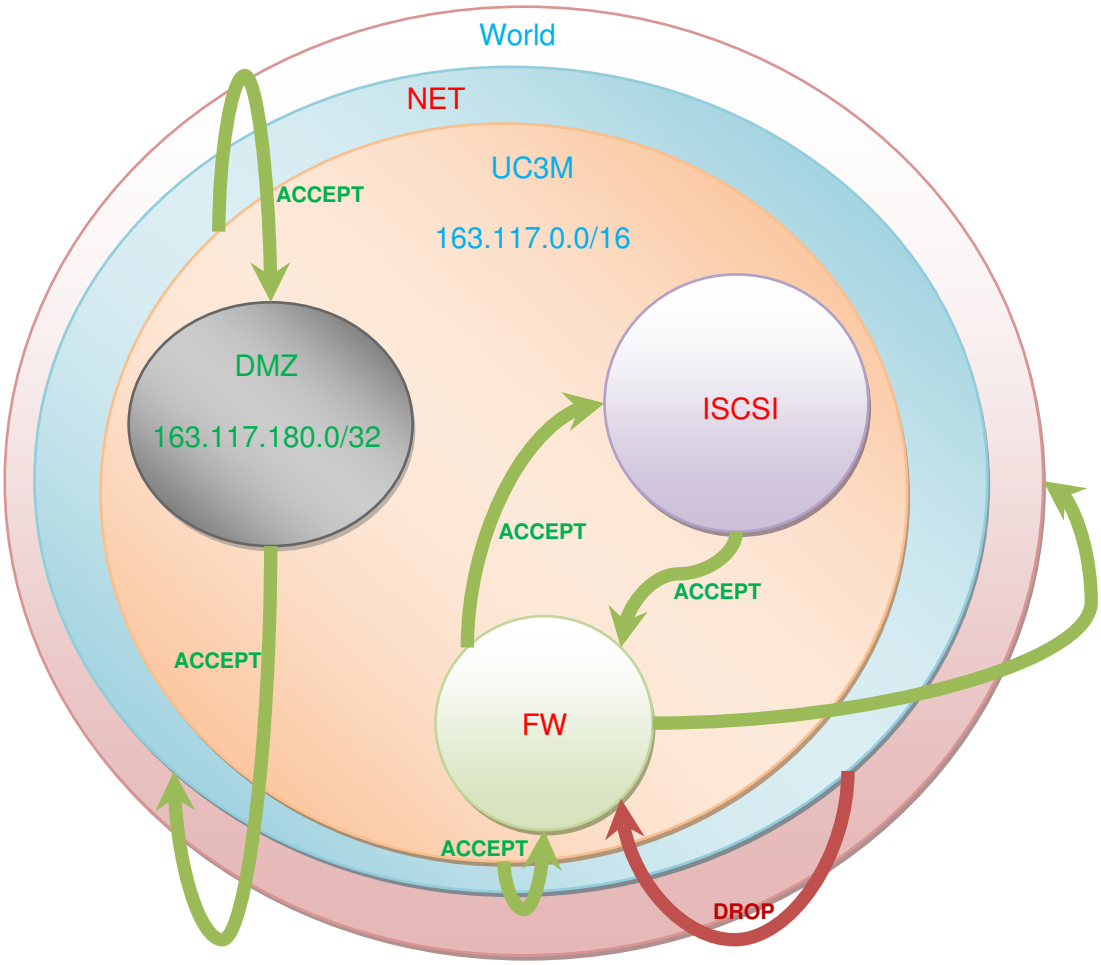

Ilustración 53 Esquema de política del firewall

4.4.2. Diseño de la funcionalidad de backup

Aunque Proxmox permite automatizar a través de su interfaz el proceso de programar las copias de seguridad de las máquinas virtuales, no se ha optado por esta opción, dado que barajamos varias opciones en cuanto a la manera de realizar el backup (completo, diferencial e incremental) que se detallarán más adelante, además

de tener que orquestar dos tipos de backup (local y corporativo) que deben hacerse en un orden concreto. Por lo tanto se prefiere realizar unos scripts que automaticen el proceso de script que lanzaremos mediante el demonio *cron* del sistema GNU/Linux. La alternativa de usar scripts propios nos ofrece una mayor potencia en cuanto a la programación de las copias y la manera de realizarlas.

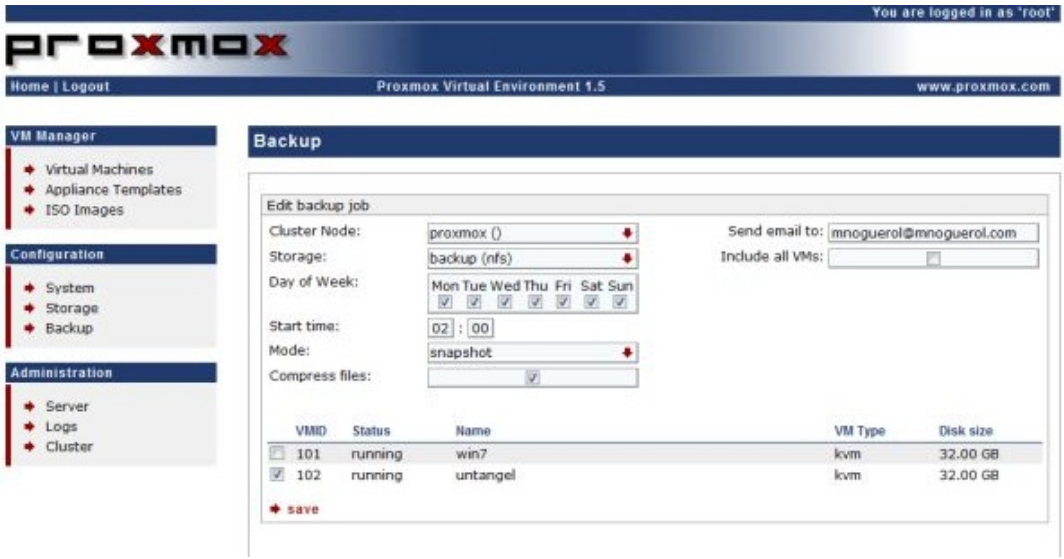

Ilustración 54 Definición del backup en Proxmox

El backup de las máquinas virtuales se almacena en el disco local del nodo del servidor de teletrabajo en el que corre la máquina virtual en cuestión. Ante cualquier problema en una máquina virtual se podría restaurar el backup que tiene almacenado el servidor de teletrabajo, con la copia de seguridad más actualizada que existe.

Sin embargo los servidores de Teletrabajo son servidores que utilizan un disco magnético como medio de almacenamiento secundario y, por tanto, el espacio del que dispone está limitado para realizar las funciones de backup. Además, se almacenarán las copias de seguridad más recientes, por ser las más interesantes para restaurar, dado que minimizan la pérdida de datos para el teletrabajador. En el proceso de detección de incidentes se hará uso del sistema de monitorización, que permite detectar rápidamente el problema.

Todo esto implica que el número de copias de seguridad que se pueden tener almacenadas en un mismo instante es pequeño, teniendo en cuenta que es imposible, restaurar las máquinas a un punto temporalmente más lejano a la última copia. Así, surge la necesidad de tener un histórico del mayor tamaño posible. El hacer uso de un servidor externo que realice las copias en cinta y con una política de preservación bien definida, ofrece una mayor seguridad quetener datos replicados en tres servidores distintos: backup externo, teletrabajo y almacenamiento iSCSI externo. Por los

motivos expuestos, en los requisitos aparece la necesidad de realizar un backup corporativo de la infraestructura, servicio que se utilizará para realizar la copia de seguridad externa en cinta.

El servicio de backup corporativo tiene como restricción (debido al gran tamaño de los datos que hay que salvar) que sólo se puede hacer una copia a la semana (los jueves a partir de las 2:00). Un backup a la semana resulta insuficiente para el servicio que se quiere dar, ya que si una máquina se estropea un miércoles, implicaría que al restaurar el backup corporativo se perdería casi una semana entera de uso de la máquina, además, habría que solicitar los datos al servicio, lo que podría retrasar el tiempo de restauración del servicio. Todo esto hace necesario la realización de un backup complementario al corporativo, aprovechando el espacio de almacenamiento del servidor de teletrabajo. En su definición se han estudiado las cuatro opciones que se muestran a continuación, con el fin de proporcionar una solución que permita: tener el máximo número de puntos de restauración en el espacio de almacenamiento del servidor de teletrabajo; generarlas en un tiempo lo suficientemente pequeño para que no afecte a la actividad de los teletrabajadores; y lograr realizar todas las copias de seguridad con la frecuencia escogida.

Para decidir cual es la mejor forma de realizar el backup, se diseñan y estudian todos los algoritmos presentados en *2.4.2 Copias de seguridad*.

Los backups se apoyarán en el uso de una herramienta llamada *Xdelta* que se encarga de comparar dos ficheros (fichero A y fichero B) de tal manera que sus diferencias generan un tercer fichero (C) tal que al aplicar el nuevo fichero sobre el primero se puede obtener el segundo (aplicado C a A se obtiene B) por mediación de la misma herramienta.

Los backups se generan mediante la herramienta *Vzdump*, su gran ventaja es que, mediante un único identificador de máquina virtual, la herramienta se encarga de hacer una imagen del disco de la máquina (sin necesidad de pararla) y empaquetar esa imagen del disco junto con el fichero de configuración que define a la máquina virtual, utilizando un formato de tipo *tar*, o *tgz* si optamos por que los comprima. Además, el empaquetado en *tar* se realiza mediante una versión propia del programa del mismo nombre, que optimiza el tamaño del backup eliminando del mismo el espacio vacío que haya en el disco duro de la máquina virtual. Como consecuencia se obtiene un ahorro del limitado espacio para almacenamiento en local.

Independientemente del formato, *tar* o *tgz*, el resultado son ficheros binarios que pueden ser comparados mediante *Xdelta*.

Puesto que las máquinas virtuales utilizan un sistema operativo XP instalado sobre una partición NTFS, es necesario poder acceder a esas particiones. Esto implica

montarlas en los servidores de Teletrabajo, teniendo que cerrar las máquinas virtuales durante el proceso de backup. Esto provoca que resulte imposible utilizar el backup a nivel de fichero *2.4.2.5 Backup por ficheros* que podría resultar óptimo en cuanto espacio de almacenamiento requerido.

Además de los algoritmos de backup comentados para reducir el tamaño de las copias de seguridad, hay que tener en cuenta que es posible aplicar una compresión mediante *gzip*. De hecho la propia herramienta *Vzdump* permite obtener el backup directamente comprimido con *gzip*. El uso de la compresión reduce el tamaño de los backups de discos duros a aproximadamente la mitad de media, tal y como se ha podido comprobar. No obstante, el tiempo para generar el backup comprimido es ligeramente superior, aunque proporcionalmente es una alternativa interesante pues sólo necesita 7 minutos adicionales para ahorrar 10 GB de almacenamiento, en contraposición a los 25 minutos que se tarda en comprimir el backup manualmente, una vez terminado el backup. El problema que presenta la compresión es que no es conveniente utilizarla al calcular la diferencia entre backups mediante *Xdelta*, ya que el algoritmo de compresión elimina la redundancia que poseía el fichero y el orden y valor de los bytes del mismo, de manera que una comparación a nivel binario de las similitudes de dos ficheros de este tipo no sería capaz de identificarlas al comparar ficheros imágenes del mismo disco. Por ello, la propia herramienta *Xdelta*, antes de comparar los backups, hace automáticamente una descompresión, la cual mediante varias pruebas se ha determinado que dura aproximadamente 18 minutos.

Los requisitos imponen que el backup de las máquinas virtuales se debe realizar antes de que comience el servicio de backup corporativo. El servicio de backup corporativo accede a una carpeta determinada de los servidores de Teletrabajo y copia todos los ficheros que encuentre ahí. Antes de que se ejecute ese servicio, hay que dejar un backup completo de cada máquina dentro de ese directorio. Por lo tanto, el backup completo de las máquinas debe terminar antes de las 2:00. Por otro lado, también según los requisitos, el backup de las máquinas se debe hacer de la forma menos perceptible posible para los teletrabajadores (que no afecte a su trabajo). Aunque utilizar la función de *snapshot* de *Vzdump* permite realizar el backup sin necesidad de apagar la máquina virtual, mediante pruebas se ha podido comprobar que el rendimiento de la máquina virtual se ve ligeramente afectado durante el momento del backup. Siendo, por tanto, el momento idóneo para realizar el backup aquel en el que el uso de las máquinas sea el mínimo. Por otro lado, dado que el backup corporativo sólo se realiza una vez por la semana, cuanto más actualizados sean los datos que se almacenen en él, más valor tendrá para los teletrabajadores en el momento de recuperarse ante un problema. De este modo, es más interesante realizar el backup después de la jornada laboral, frente a realizarlo antes de la jornada, ya que

minimiza el lapso de tiempo que separa la última sesión de trabajo salvada y el almacenamiento en cinta del backup corporativo. Si tenemos en cuenta que la jornada no se prolongará más allá de las 21:00, obtenemos que la horquilla de tiempo en la que podemos realizar el backup está comprendida entre las 21:00 y las 2:00, dándonos 5 horas para realizar el backup de las 15 máquinas virtuales.

En la *Tabla 139 Resumen de los parámetros del backup* se muestra un resumen de los parámetros y restricciones que tenemos en cuento a los servidores de Teletrabajo y las mediciones que se han realizado mediante pruebas hechas con los distintos algoritmos explicados.

	Valor
Almacenamiento disponible para backup por nodo	330 GB
Almacenamiento disponible para backup	660 GB
Disco duro de máquina virtual	20 GB
Tiempo disponible para backup	5 horas
Tiempo de backup de una máquina	18 minutos
Tiempo de backup de una máquina comprimida	25 minutos
Nº máximo de máquinas virtuales por nodo	8
Nº total de máquinas virtuales	15
Tiempo de descompresión	20 minutos
Tasa de compresión media de backup completo	0,48
Tasa de compresión de backup diferencial	0,11
Tiempo de compresión	25 minutos
Tiempo de cálculo de la diferencia	12 minutos
Tiempo de restauración de backup diferencial	29 minutos
Tamaño medio de backup diferencial	300 MB

Tabla 139 Resumen de los parámetros del backup

Si cada backup ocupa 20 GB, hay 330 GB disponibles en cada servidor de teletrabajo, y el número máximo de máquinas en un servidor es 8, significa que se pueden almacenar a la vez 2,025 backups completos para cada máquina en cada nodo. Si se llegasen a hacer dos backups completos, la partición quedaría tan llena que sería imposible poder realizar operaciones adicionales que necesiten de un fichero auxiliar. El poco espacio del que se dispone hace imprescindible tener que recurrir a la compresión para los backups. No sólo eso, a pesar de que las pruebas realizadas con backups incrementales y diferenciales, en media, dan resultados muy buenos en cuento a espacio ocupado, se ha detectado que los cambios realizados en las máquinas no tienen una correspondencia directa en el tamaño que ocupará su diferencia. Por ejemplo, una instalación en la máquina virtual de unos 600 MB tuvo como resultado un backup diferencial de 2 GB, lo que lleva a pensar que hacer una comparación binaria entre dos discos duros no tiene un comportamiento deseable, porque la posición física en bloques de los datos del disco no parece muy estable en función del tiempo. Esto nos lleva al problema de que un cambio grande en la máquina virtual puede provocar cambios en casi la totalidad del disco duro, haciendo que los backups diferenciales sean tan grandes como los completos. Esta situación puede ser grave si se produce un cambio que afecte a todas las máquinas a la vez, como puede ser una actualización del sistema o del software que tienen instalado, pudiendo llegar a no caber los backups en el disco local de Teletrabajo.

Si se utiliza la opción *--compress* de *Vzdump* conseguimos, como se ha mencionado anteriormente que el backup completo ocupe 11 GB de media, pudiendo tener al menos 3,75 backups completos de cada máquina a la vez. Realizar un backup completo utilizando la compresión, si tenemos en cuenta que en el peor de los casos hay que realizar el backup de 8 máquinas y cada backup dura 25 minutos, tardaría 3 horas y 20 minutos, inferior a las 5 horas máximas, lo que proporciona margen de tiempo suficiente para realizar el backup corporativo.

Según los datos de la *Tabla 139 Resumen de los parámetros del backup* si se aplica el backup diferencial se tiene un tiempo máximo para realizar el backup de 50 minutos (tiempo de realizar el backup + tiempo para descomprimir el de referencia + tiempo de cálculo de diferencia), teniendo en cuenta que el servidor contiene un máximo de 8 máquinas, da un tiempo total teórico de backup de 6 horas y 40 minutos. No obstante, una vez implementado el script y realizadas diversas pruebas de backup diferencial en una situación real, se obtienen resultados muy dispares, empleando tiempos para realizar el backup que van desde las casi 7 horas teóricas, hasta sobrepasar las 12 horas, lo que incumple el requisito definido en *Tabla 124: Requisito del software RSNF-18,* si consideramos que el teletrabajador puede comenzar a trabajar a las 9 de la mañana. Una explicación para esta desviación en el

tiempo de los cálculos teóricos es la observación del incremento del tamaño en los ficheros de backup diferenciales hasta 2 GB, requiriendo un mayor tiempo de escritura en disco de cada backup, lo cual supone un problema al haber detectado la velocidad de escritura en el disco local como el cuello de botella en las operaciones de backup.

Dados los malos resultados del backup diferencial, ni se plantea utilizar el backup incremental, pues su funcionamiento requiere muchas más escrituras en disco duro para obtener los backups temporales.

Finalmente se ha optado por la solución más sencilla y robusta, la de realizar únicamente 2 backups completos a la semana: uno el miércoles y otro el domingo, perdiendo de esta manera 5 puntos de restauración a la semana. Aunque no sea la mejor solución, sí es la única con la que se asegura cumplir los requisitos establecidos. Quizás en un futuro, si se dispone de un dispositivo de almacenamiento de mayor capacidad y velocidad (como otro disco en red iSCSI) se pueda cambiar la política de backup, por lo que se mantendrá esta funcionalidad en los scripts.

El último paso para completar el diseño del backup es lo que denominamos backup cruzado. El objetivo de este tipo de backup es realizar una copia de seguridad, lo más actualizada posible, de la configuración de todas las máquinas en todos los nodos del clúster de Teletrabajo. De esta manera, ante un fallo en cualquiera de los nodos que lo deje sin funcionamiento, es posible restaurar las máquinas virtuales que daban servicio en ese nodo en otro nodo distinto. En el peor de los casos no se tendría acceso al servidor caído, no pudiendo recuperar las máquinas virtuales que hospeda para poder restaurarlas en otro servidor. Esto hace imprescindible que se realice este backup con la mayor frecuencia posible.

La realización del backup cruzado prodra hacerse a partir del backup completo generado en el paso previo. Copiar los backups completos realizados durante la operación de backup local presenta un gran número de inconvenientes. El principal inconveniente es que siempre existe un periodo de tiempo (ya sea grande o pequeño) entre que se realizó el backup y se restaura en otro servidor de Teletrabajo, ese espacio de tiempo puede conllevar una pérdida de trabajo, que hace más interesante al teletrabajador la opción de esperar a que se restaure el sistema para retomar su trabajo en lugar de tener que rehacerlo, perdiendo entonces el sentido de este tipo de backup. Aunque ese problema no puede ser solventado, si se incrementa la frecuencia con la que se realiza el backup podría ser minimizando. Según las limitaciones del backup local que se ha diseñado, la frecuencia máxima que se podría establecer para realizar estos backups sería de 3 horas y 20 minutos, implicando que los servidores de teletrabajo estarían continuamente trabajando en funciones de backup, además se estarían realizando mientras los teletrabajadores trabajan con sus máquinas virtuales,

lo que afectaría a su rendimiento. Finalmente, sería necesario reservar en cada servidor de teletrabajo al menos 20 GB de almacenamiento para cada máquina virtual del resto de nodos, que en el peor de los casos serían 8 máquinas virtuales, es decir 160 GB, que dados los problemas con el reducido almacenamiento del que disponemos para realizar los backups, y que de hecho ha limitado mucho el número de backups realizados, sería necesario ampliar el almacenamiento de los servidores para llevar a cabo esta tarea, o reducir a 1 el número de backups locales que se almacenarían en los servidores.

La opción de realizar backups convencionales es inviable debido a las limitaciones de las máquinas expuestas en el párrafo más atrás. No obstante recordemos que como se explica en el documento con la primera parte de este proyecto (Gil Bázquez, 2011), los discos duros de las máquinas virtuales están almacenados en un servidor iSCSI que los servidores de Teletrabajo acceden mediante una red local. El acceso por red a los discos duros hace que los mismos sean visibles por todos los nodos de teletrabajo, por lo que sería factible arrancar los discos duros de las máquinas virtuales de cualquier otro nodo, simplemente sería necesario que todos los servidores conociesen las especificaciones de cada una de las máquinas. En "*/etc/qemu-server*" se pueden encontrar los ficheros que definen las máquinas virtuales (identificador de su disco duro, identificador de máquina, dirección MAC y dispositivos conectados a ella), con el fichero que define la máquina virtual y el disco que utiliza es posible arrancar dicha máquina virtual en cualquier servidor de Proxmox. Los ficheros de configuración son ficheros de texto plano muy pequeños (unos 180 bytes), haciendo muy sencillo su replicación, transmisión y almacenaje en los nodos del servidor de Teletrabajo. El backup cruzado consistirá en una conexión *ssh* que se realizará al final del proceso de backup local con el resto de máquinas del clúster, mediante la cual se envían los ficheros de configuración para almacenarlos en el directorio "*/etc/qemu-server/*NOMBRE-ORIGEN", siendo NOMBRE-ORIGEN el nombre definido para la máquina anfitriona original de esas máquinas.

4.4.3. Diseño de la función de monitorización

El servidor de monitorización es creado como parte de este proyecto para supervisar y emitir alertas sobre todas las máquinas implicadas en el Servicio de Teletrabajo (incluido el propio servidor de monitorización). No obstante, no será el único uso que se le dará, estando planificada la monitorización de los servidores de la Oficina de Software Libre de la Universidad Carlos III. También está previsto que el

servidor de Teletrabajo hospede otro tipo distinto de máquinas virtuales que ofrezcan otros servicios. Por todo esto, resulta conveniente realizar un buen diseño de plantillas y jerarquía de las mismas, que permita simplificar el proceso de monitorizar nuevas máquinas virtuales aplicando únicamente la plantilla deseada. Además, si se aprovecha la herencia en las plantillas, se simplificará el proceso de ajustar parámetros de monitorización, eliminar algunos innecesarios o añadir nueva funcionalidad en todas las plantillas específicas y, por tanto, en todas las máquinas en las que son utilizadas.

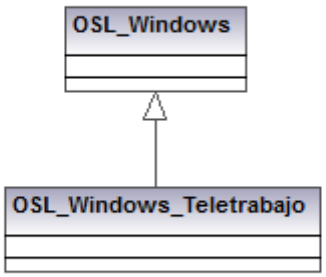

Ilustración 55 Jerarquía de plantillas Windows

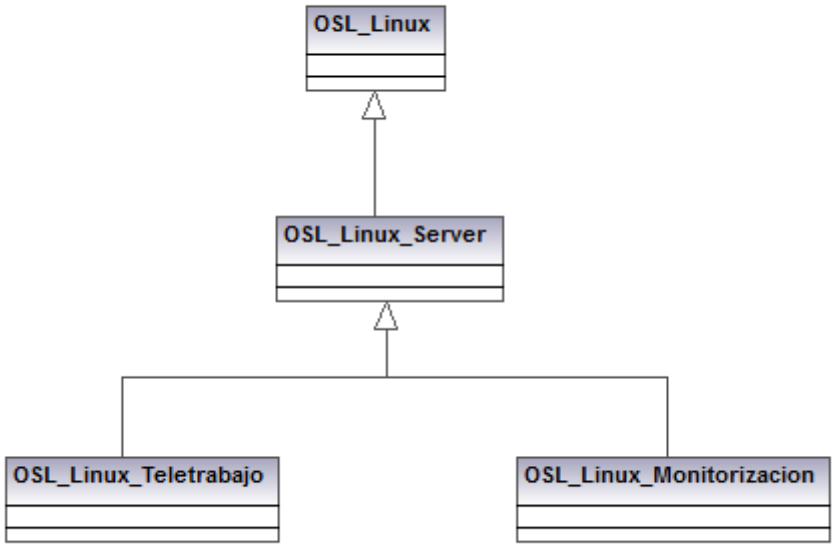

Ilustración 56 Jerarquía de plantillas Linux

Se diseñan seis plantillas en dos árboles distintos, uno para sistemas Windows y otro para sistemas Linux. El motivo por el que es necesario tener un árbol distinto para cada sistema operativo es que la configuración para monitorizar ambos sistemas es distinta, en especial en cuanto a mediciones de dispositivos y en cuanto a los parámetros que permite medir cada sistema.

La plantilla de Windows contiene todas las mediciones que se pueden realizar en una máquina virtualizada en Proxmox, con el sistema operativo Microsoft Windows instalado. La restricción de que sea virtualizada en Proxmox es importante, ya que es necesario conocer algunos detalles sobre los controladores de dispositivos instalados en los sistemas, dado que la plantilla está diseñada para un hardware específico. Esta plantilla monitoriza elementos básicos del hardware virtualizado como uso de CPU o memoria (los gráficos mostrarán estos mismos elementos básicos), algunos parámetros como son la temperatura de la CPU son eliminados, pues no tienen sentido en equipos virtuales. Las alertas especificadas, puesto que se trata de una máquina Windows genérica, no deberían tener valores concretos numéricos en sus condiciones, recurriendo al uso de porcentajes para definir las alertas en la medida de lo posible. También por ser una máquina genérica, no se posee especial interés en el estado de los elementos monitorizados, excepto que la máquina este inaccesible, por ello es la única notificación que se definirá para esas máquinas.

Hay que destacar el elemento monitorizado denominado *interfaces_red* el cual nos da el nombre que tiene en Windows la interfaz de red. El nombre de la interfaz de red es necesario conocerlo para la construcción del elemento monitorizado que de la información sobre el tráfico entrante y de salida de la máquina. A diferencia de los equipos Linux (en los que mediante el comando *ifconfig* podemos obtener los nombres de los interfaces de red, y que además típicamente tienen nombres como eth0, eth1, wlan, loop etc.), el interfaz de red en Windows tiene un nombre difícil de consultar en el propio sistema, ya que utiliza el nombre del controlador del dispositivo con su nombre en inglés, independientemente de que Windows únicamente nos da su nombre en castellano.

Se puede observar que muchos de los parámetros monitorizados pueden llegar a ser redundantes, de manera que algunos pueden ser calculados a partir de otros (por ejemplo el porcentaje de disco duro libre se puede calcular a partir del tamaño total del disco duro y el espacio libre del disco duro). Hacer que los equipos monitoricen estos elementos redundantes y los envíen al servidor central de monitorización no supone una carga computacional adicional para esos equipos. En cambio, realizar estos cálculos cada pocos segundos, y para un gran número de equipos, sí supondría una carga adicional para el servidor de monitorización, lo que podría provocar un retardo en las inserciones de valores de elementos en la base de datos.

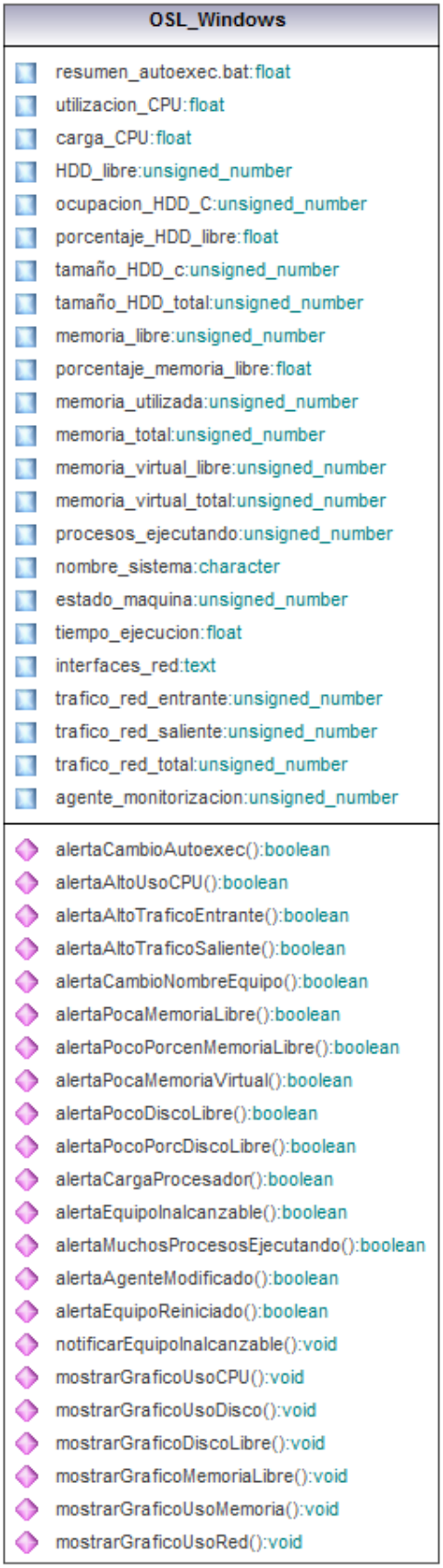

Ilustración 57 Plantilla OSL Windows

A partir de la plantilla OSL Windows para máquinas Windows genéricas se define OSL Teletrabajo, la cual hereda de OSL Windows y añade la funcionalidad de monitorización específica para las máquinas de teletrabajo. La funcionalidad principal que se tiene que añadir es la monitorización del servicio RDP que ofrece Windows, enviando una notificación cuando el servicio deja de funcionar. La notificación y alerta sobre el estado del servicio RDP es recomendable hacerla dependiente de la alerta que indica que el equipo es inalcanzable, ya que, si el sistema no está disponible, todos los servicios que ofrezca tampoco estarán disponibles. Mediante la dependencia logramos que siempre que salte una alarma, todas las alarmas dependientes de ella quedan inhibidas, logrando así que únicamente se mande una notificación: "equipo inalcanzable" en lugar de "equipo inalcanzable" y "RDP inactivo".

Finalmente, la plantilla para las máquinas de teletrabajo redefine la notificación que se da cuando el equipo esta caído. Durante la fase de producción del Servicio de Teletrabajo se detectó que, regularmente (una vez a la semana), saltaban las alertas de todas las máquinas virtuales de teletrabajo, indicando que esas máquinas estaban caídas y enviando una notificación por correo electrónico. Sin embargo, se pudo comprobar que las máquinas seguían funcionando correctamente, a través de la consola que ofrece Proxmox para visualizar las máquinas virtuales. Este comportamiento se asocia con las actualizaciones del sistema operativo, que interrumpen momentáneamente la conectividad. Gracias a las estadísticas históricas que ofrece el sistema de monitorización se ha podido comprobar el tiempo medio de inactividad de las máquinas durante estas actualizaciones, comprobando que en ningún momento supera los 8 minutos. Por lo tanto, para evitar falsas alarmas con la consiguiente notificación, se ha redefinido la notificación de manera que sólo sea enviada si la máquina virtual esta al menos 8 minutos inactiva.

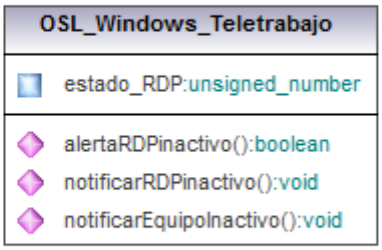

Ilustración 58 Plantilla OSL Windows Teletrabajo

La plantilla de Linux realiza las mismas comprobaciones que la de Windows: uso de CPU, consumo de memoria, disco duro libre, etc. Puesto que los sistemas GNU/Linux suelen tener varias particiones, esta plantilla mide el uso de las particiones "/" y "/home" que suelen tenerse tanto en equipos de escritorio como en servidores. Aunque se mida el uso de ambas particiones, no tiene por qué ser necesario que existan ambas en el equipo monitorizado. El elemento monitorizado no medirá parámetros de la partición en sí, sino la partición a la que pertenece el directorio indicado. Por tanto si suponemos que el equipo únicamente tiene una partición, la medición de "/home" nos dará los resultados mismos resultados que la medición en "/".

El agente de Linux permite conocer mucha más información de la que proporciona el agente de Windows, como es el porcentaje de cada uso de la CPU (usuario, sistema, ociosa, bloqueada…) que, aunque para los objetivos que tenemos de monitorización no son muy relevantes (ya que son datos muy difíciles de interpretar para detectar una anomalía), si son útiles si se representan en una gráfica para verificar la estabilidad en el uso de CPU de los servidores.

Los elementos monitorizados "resumen" aplicarán una función resumen de los archivos indicados. Las alarmas de resumen compararán el último valor obtenido de los elementos monitorizados resumen y, si es distinto a los almacenados anteriormente, se activarán. Mediante las alertas de resumen se puede detectar si se ha realizado algún cambio no autorizado en alguno de los ficheros del sistema. La comprobación del resumen se hace sobre los directorios:

- "/vmlinuz" (donde se almacenan los núcleos disponibles en determinados sistemas GNU/Linux).

- "/etc/passwd" (define los usuarios del sistema, el grupo al que pertenecen, sus identificadores, home y shell).
- "usr/bin/ssh" y "/usr/sbin/sshd" (verifica que el programa cliente de ssh y el demonio de servicio no han sido comprometidos por un rootkit).
- "/etc/services" (almacena los servicios que conoce el equipo y la dirección y/o puerto en el que puede ser localizado).

Otra diferencia, respecto a la plantilla de Windows, es que en Linux se puede contar el número de procesos que hay en ejecución con un determinado nombre, permitiendo saber si se está ejecutando un determinado proceso (el número de procesos con ese nombre es mayor que 0). Así se comprueba qué procesos importantes, como el demonio de ssh que permite el acceso remoto al equipo, están funcionando, o que está funcionando el proceso de log "syslog" del sistema.

Una de las alertas más importantes, que de hecho cuenta con una notificación, es el estado del firewall en el equipo monitorizado. En caso de que el firewall dejase de funcionar, se mandaría una notificación para que el administrador pudiese volver a levantarlo en el menor tiempo posible, o incluso que pueda levantarlo automáticamente el servidor de monitorización. El problema se presenta en cómo detectar que el firewall está activo o no. Como se ha explicado en el párrafo anterior, es posible conocer si un proceso está en ejecución o no, por lo que bastaría con crear un elemento monitorizado que contase el número de procesos "firewall" que están ejecutándose en el momento. El firewall de los servidores de Teletrabajo, está configurado mediante *Shorewall* (*3.2.1.1 Shorewall*), pero *Shorewall* no es un demonio que este continuamente en ejecución, se ejecuta una vez y deja configurado *Netfilter*, haciendo imposible conocer el estado del firewall a través de *Shorewall*. Por su parte *Netfilter/iptables* tampoco es un demonio, sino un framework, imposibilitando también conocer su estado contando procesos. Existen otras dos opciones para conocer un estado del equipo: la primera es recurrir a un fichero de log, costoso de analizar analizar para obtener el dato que se desea; y la segunda, ejecutar un comando que dé como respuesta el valor que se desea conocer. No hay ningún comando que devuelva un valor que nos permita conocer el estado del firewall directamente, por una parte existe el comando *iptables*, el cual, mediante la opción *–L* muestra por pantalla todas las cadenas y reglas definidas en esas cadenas. La información que da este comando presenta los mismos problemas que el log, es difícil de interpretar. La solución propuesta pasa por generar un script de *bash* que invoque el comando *iptables* y compruebe si existen reglas definidas en las cadenas por defecto (INPUT, OUTPUT y FORWARD). Si existe alguna regla el script devolverá un valor numérico mayor que '0'.

El problema que presenta la solución del script es que deberá distribuirse en todas las máquinas Linux monitorizadas y ser almacenado en un mismo directorio en todas ellas, de manera que se pueda utilizar en todos los equipos el mismo elemento monitorizado definido en la plantilla. Adicionalmente, puesto que el comando *iptables* sólo puede ser ejecutado por *root* y el script será ejecutado por el agente de monitorización, habrá que hacer que el agente de monitorización tenga potestad para consultar iptables.

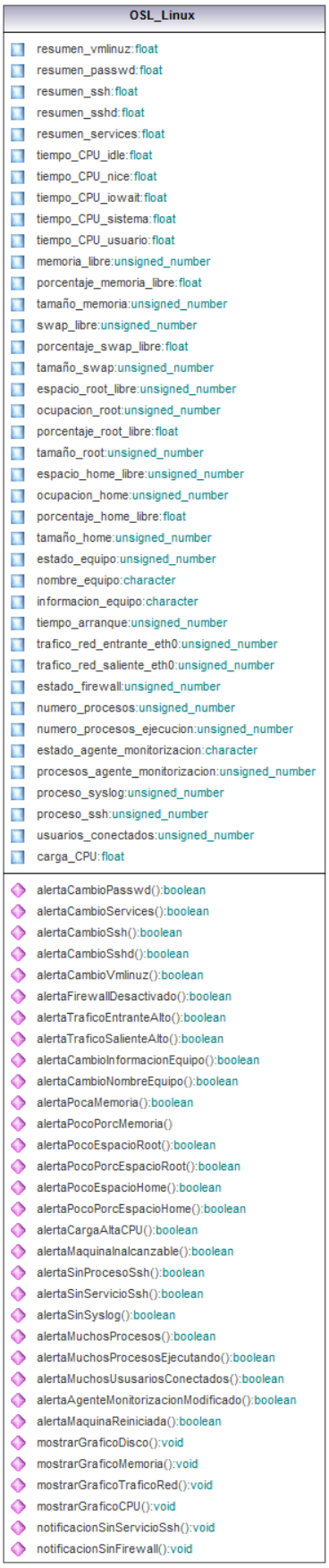

Ilustración 59 Plantilla OSL Linux

Se extiende la funcionalidad que aporta la plantilla de Linux para tener en cuenta los elementos adicionales que presentan los "típicos" servidores de Linux, se acaban obteniendo suficientes nuevos elementos para definir una nueva plantilla: OSL Linux Server. La filosofía de esta plantilla es distinta a la de las anteriores, ya que en las otras se tenían alertas presentes en todos los equipos a los que estaban destinadas las plantillas. Con OSL Linux Server se monitorizan todos los servicios típicos en servidores. Puesto que no todos los servidores tienen por qué ofrecer todos los servicios, activar la plantilla completa puede generar alertas continuamente activadas indicando el fallo de servicios que no ofrece realmente el servidor. Para evitar esas falsas alarmas, en el momento que se aplique esta plantilla en el servidor, el administrador tendrá que desactivar todas las alertas no deseadas.

Puesto que los servidores Unix suelen tener varias particiones específicas para cada uno de los directorios del sistema, esta plantilla añade la monitorización del espacio disponible en las particiones "/opt", "/var", "/tmp" y "/usr" que se añaden a "/home" y "/" heredados de la plantilla OSL Linux.

En la plantilla se puede observar que se monitorizan los servicios de dos formas distintas: verificando que está en ejecución el proceso que ofrece ese servicio, y accediendo directamente al servicio mediante una conexión TCP o UDP. La redundancia en la monitorización de servicios permite acotar las posibles causas de un problema, en caso de que se obtenga una alerta por caída de un servicio, pudiendo así discriminar si es un problema de conectividad, o del propio proceso.

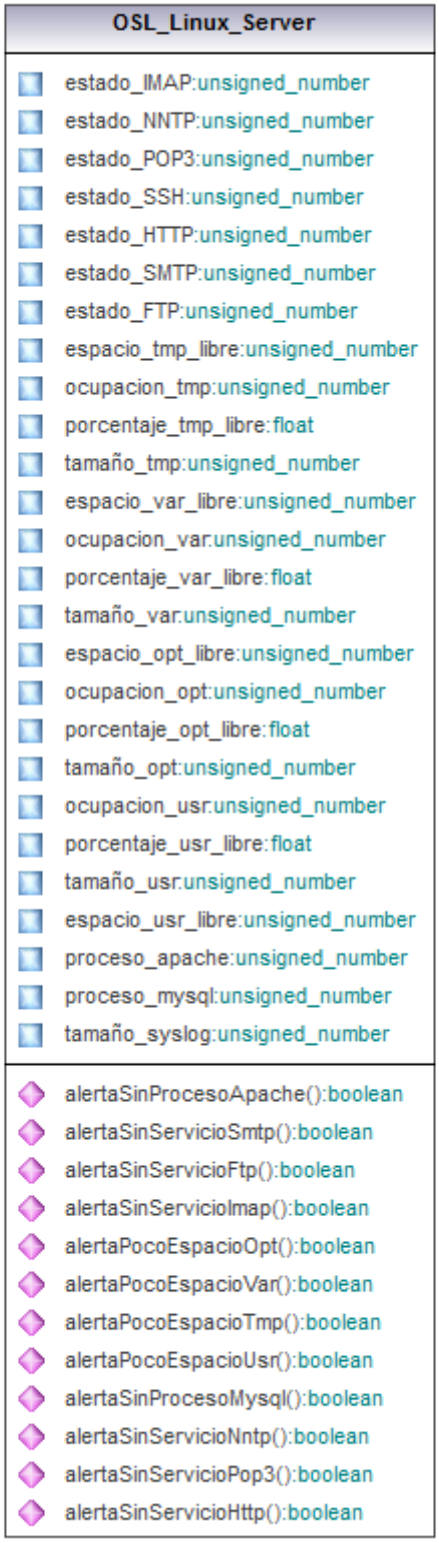

Ilustración 60 Plantilla OSL Linux Server

La plantilla utilizada para el servidor de monitorización únicamente añade dos elementos monitorizados nuevos y una única alerta. Por un lado se comprueba que

está funcionando el proceso que realiza la función de recopilar los datos de los agentes e introducirlos en la base de datos, saltando la alerta cuando deje de funcionar; por otro, se obtiene información sobre el servicio de monitorización en ejecución, como puede ser su versión.

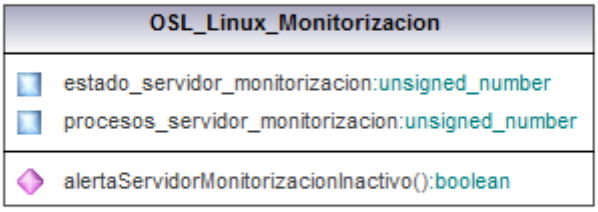

Ilustración 61 Plantilla OSL Linux Monitorización

La última plantilla diseñada es la que se aplica en los servidores del Servicio de Teletrabajo. Como los servidores de Teletrabajo poseen dos interfaces de red físicos, la nueva plantilla contempla el nuevo interfaz de red, redefiniendo las alertas de entrada y salida de tráfico para tener en cuenta la nueva interfaz. También se redefinen las alertas que controlan el número de procesos en ejecución, dado que el número de procesos que ejecuta un servidor de Proxmox es muy elevado, y proporcional al número de máquinas virtuales que hospeda. Es preciso ajustar el límite en el que se activa esta alerta mediante prueba y error, según el número máquinas virtuales que hospede.

Ilustración 62 Plantilla OSL Linux Server Teletrabajo

4.5. Implantación del sistema

Completada la fase de diseñose han definido todos los aspectos que se deberán desarrollar para la construcción del sistema. Como se mencionó en el punto *4.1 Fase inicial*, cuando se escogió el ciclo de vida software para el desarrollo del sistema, se escogió el ciclo de vida en cascada, el cual definía después de la fase de diseño una fase denominada codificación. Dado que la codificación en sí es mínima en este proyecto, limitándose a la escritura varios scripts en Bash, se ha decidido renombrar esta fase de "codificación" como "implantación del sistema".

Esta fase se iniciará durante la fase de implantación del proyecto anterior del servidor de Teletrabajo, estando ya instalado el sistema operativo y la funcionalidad de virtualización, se pasará a implementar la funcionalidad de control de acceso y backup, que son las características más importantes e imprescindibles para poder poner en producción el Servicio de Teletrabajo.

Una vez finalizada la configuración del servidor de Teletrabajo se pasará al desarrollo del servicio de monitorización, ya que es en este punto cuando se empiezan a tener máquinas virtuales funcionales que poder comenzar a monitorizar y con las que hacer las pruebas de monitorización. Además, es en este punto en el que está implementada toda la funcionalidad de los servidores de Teletrabajo para poder medir la actividad de esos servicios, permitiendo probar la funcionalidad de alertas y notificaciones según estén activos o no dichos servicios.

4.5.1. Configuración del control de acceso

Como se ha explicado durante el punto *4.3 Diseño arquitectónico*, la herramienta utilizada para el control de acceso será *Shorewall*. Los servidores de Teletrabajo funcionan con un sistema operativo que simplifica el despliegue de la infraestructura de virtualización denominado Proxmox (*3.1.1 Plataforma de virtualización: Proxmox*). Proxmox es una distribución de GNU/Linux basada en Debian, por lo que contamos con las herramientas que se distribuyen con Debian, en concreto, su gestor de paquetes *Advance Package Tool* (o APT), por lo que la instalación de *Shorewall* se realizará utilizando esta herramienta.

En primer lugar, se accede a todos los servidores de Teletrabajo mediante ssh y se introduce en todas las máquinas el siguiente comando:

```
sudo apt-get install shorewall
```

Mediante el comando sudo, indicamos al usuario "root" que ejecute un comando que el usuario normal no tiene privilegios para ejecutar. Para que esto funcione, durante la instalación del sistema operativo, o mediante la modificación del fichero "sudoers" indicamos que el usuario tiene privilegios para invocar el comando sudo.

Tras ejecutar el comando solicitará la contraseña de usuario, que al introducirla comenzará la descarga del paquete shorewall y lo instalará. A continuación comprobamos que la instalación ha creado un directorio "/etc/shorewall/" que contenga al menos los ficheros que permitan la compilación de la definición del firewall, estos ficheros son "Makefile" y "shorewall.conf".

Para registrar en un fichero de log todos los intentos de conexión que han sido cortadas por el firewall, editamos el fichero "shorewall.conf" con la ayuda del editor de textos vim:

```
cd /etc/shorewall
sudo vim shorewall.conf
```

En el fichero que se muestra en la pantalla se busca una línea que comience con el texto "LOGFILE" y se indica en esa línea donde queremos que almacene el log, que será en el directorio "/var/log/shorewall/":

```
LOGFILE=/var/log/shorewall
```

También en este fichero hay que indicar si queremos arrancar el firewall mediante shorewall manualmente o si queremos que se haga automáticamente al encender el equipo. Dado que sí queremos que se inicie automáticamente para ofrecer una mayor seguridad (por ejemplo un arranque debido a un corte del suministro eléctrico o un descuido del administrador), se modifica la línea STARTUP_ENABLED:

```
STARTUP_ENABLED=Yes
```

Se salvan los cambios y se cierra el fichero para a continuación crear los cinco ficheros de configuración necesarios para compilar *Shorewall*:

```
touch interfaces rules zones params policy
```

En estos cinco ficheros es donde quedará definido el diseño de firewall explicado en el punto *4.4.1 Diseño del control de acceso*. En primer lugar hay que definir las zonas en las que se dividió la red, para lo que hay que editar el fichero **zones** añadiendo lo siguiente:

```
#ZONE           TYPE    OPTIONS
fw              firewall
world           ipv4
net:world       bport
```

```
dmz:world       bport
iscsi           ipv4
```

La zona “fw” que representa la propia máquina se identifica mediante “firewall”, mientras que las zonas “iscsi” y “world” son las que están conectadas directamente por cable Ethernet utilizando el protocolo IP versión 4 (ipv4). Por último las zonas “net” y “dmz” son definidas como subconjuntos de la zona “world”, y se conectan mediante puertos del bridge virtual del servidor de Teletrabajo (bport).

Una vez definidas las zonas, se indica en el fichero **interfaces** a cuál de las zonas definidas se corresponden las interfaces de red del servidor:

```
#ZONE   INTERFACE       BROADCAST       OPTIONS
world   vmbr0           detect          bridge
net     vmbr0:eth0.57
dmz     vmbr0:tap+
iscsi   eth1            detect
```

Puesto que la cabina de almacenamiento en red iSCSI está conectada a través del interfaz eth1, no hace falta más que corresponder esa interfaz con la zona “iscsi”. En cambio el resto de zonas se definen a través del bridge virtual, que es quien está conectado a la interfaz eth0. El bridge está definido como vmbr0 y todo el tráfico que venga a través de él es lo considerado como la zona “world”. Dentro de la zona “world” es donde se encuentra la subzona *dmz* que es donde están conectadas las máquinas virtuales, habiendo un puerto del bridge distinto conectado a cada una de las máquinas virtuales. Todas las interfaces de las máquinas virtuales se generan dinámicamente con nombre tap{ID_máquina}i0d0, dado que inicialmente no conocemos todas las máquinas virtuales que habrá y no queremos modificar la configuración del firewall cada vez que se crea una nueva máquina, se utiliza la sentencia “tap+” para indicar todas las interfaces que contienen ese nombre (permitiendo tratar todas las máquinas como una única zona). El último puerto del enrutador es el conectado a la interfaz de red eth0, a través del cual se tiene el acceso a la red externa al servidor de Teletrabajo, es decir “net” (el nombre de puerto eth0.57 es debido a que hay configurada una vlan para el tráfico de las máquinas virtuales).

Ni en el fichero zones, ni interfaces, se hace ninguna referencia a otra de las zonas definidas durante el diseño que es la red interna de la Universidad denominada “UC3M”. Puesto que esta red está contenida dentro de la zona “net” y no hay ningún interfaz o puerto del bridge conectado directamente a ella, siendo identificada únicamente por la subred a la que pertenecen sus direcciones IP. Por estos motivos

esta zona se define dentro del fichero **params**, que permite crear variables que identifican equipos o redes.

```
SUBNETUC3M=163.117.0.0/16
```

Definidas las zonas parámetros e interfaces ya se puede comenzar a restringir o permitir tráfico mediante políticas, y las excepciones a esas políticas mediante reglas. Primero se definen las políticas en el fichero **policy**:

```
#SOURCE         DEST            POLICY          LOG LEVEL
#A Internet
dmz             net             ACCEPT
$FW             world           ACCEPT

#A Firewall
iscsi           $FW             ACCEPT
net             $FW             DROP            info

#A Almacenamiento
$FW             iscsi           ACCEPT

# THE FOLLOWING POLICY MUST BE LAST
all             all             REJECT          info
```

Para las políticas que implican cortar conexiones, se indica mediante la palabra info que queremos que las conexiones interceptadas sean registradas en el log pero únicamente con una importancia informativa.

Por último se edita el fichero **rules** especificando que tráfico de la zona net queremos permitir, en lugar de ser tirado:

```
#ACTION         SOURCE          DEST            PROTO   DEST
SOURCE          ORIGINAL        RATE            USER/   MARK
#                                                       PORT
PORT(S)         DEST            LIMIT           GROUP


# Cualquier trafico en la UC3M
ACCEPT          net:$SUBNETUC3M         $FW #Se permite el
trafico desde la Universidad al Firewall
ACCEPT          net:$SUBNETUC3M         dmz #Se permite el
trafico desde la Universidad a DMZ


# Accept SSH connections for administration
```

```
    #
    SSH(ACCEPT)      net:$SUBNETUC3M          $FW #Se permite el
trafico SSH desde la Universidad al Firewall (garantiza acceso si
se elimina la anterior
```

En el fichero se ha definido la política de acceso que implicaba a la zona diseñada "UC3M" pero, puesto que realmente la red de la Universidad se ha definido como un parámetro, no debía ser incluido en el fichero policy.

Además, como medida de seguridad por si se elimina la regla que permite el acceso desde la Universidad al equipo, se permite el tráfico ssh desde la Universidad hasta la máquina. Como se ha dicho, sólo se deja esta regla por seguridad ya que, al ser un caso particular de las anteriores, queda cubierta por ellas y no se aplica.

4.5.2. Configuración del backup

Según el diseño, hay tres tipos de backup distintos: normal, corporativo y cruzado. Esta funcionalidad sí será codificada en scripts de bash: un script de control, otro que realice el backup cruzado y otro que haga el backup normal y corporativo, ya que ambas son operaciones muy similares.

El script que realiza el backup normal permite automatizar dos tipos de operaciones: crear backups y restaurar backups.

```
crisol@prox1:~$ /usr/local/bin/vmtools/vmbackup
Este script realiza una copia de seguridad de las maquinas virtuales indicadas con el flag -c o restaura
 una copia de seguridad concreta para alguna maquina virtual dada con el flag -r.

Uso: ./vmbackup [-c <NUM> <DIAS> <MAQ_INICIO> <MAQ_FIN> | -r [<VMID> [<BACKUP>]]]
     -c: Genera una copia de seguridad para cada maquina virtual.
         <NUM>: Numero de copias de seguridad soportadas.
         <DIAS>: Numero de dias que se almacenaran las copias obsoletas de una maquina.
         <MAQ_INICIO>: Numero identificador de la primera maquina de la que se realizara copia.
         <MAQ_FIN>: Numero identificador de la ultima maquina de la que se realizara copia.
     -r: Restaura la copia de seguridad <BACKUP> para la maquina <VMID>. ATENCION: La maquina <VMID> ser
a parada antes de restaurar la copia de seguridad y se arrancara cuando el proceso haya terminado.
         <VMID>: Identificador de la maquina que se quiere restaurar.
         <BACKUP>: Nombre de la copia de seguridad a restaurar.
```

Ilustración 63 Uso del script de backup de máquinas virtuales

La función de creación de backups accede al directorio donde se almacenan las máquinas virtuales ("/etc/qemu-server/") para saber que máquinas hay y obtener los identificadores de las mismas. Con estos identificadores se invoca el comando *vzdump*, pasando los identificadores en un bucle, para obtener un archivo comprimido con extensión tar.gz por cada máquina virtual. El script funcionará con rangos de máquina (<MAQ_INICIO>, <MAQ_FIN>), por ello se recurre al uso de bucles. Para invocarlo habrá que indicar cuál es la primera máquina de la que se quiere hacer

backup y cuál es la última, ignorando el script el resto de identificadores fuera de este rango. Además, se le indicará cual es el número de copias que se quieren almacenar por cada máquina virtual <NUM>, de tal forma que el script, una vez finalizado el proceso de backup de una máquina virtual, comprueba el directorio donde se almacena el backup de cada máquina. Si en ese directorio hay un número de backups superior al pasado como parámetro, se eliminan los backups con fecha más antigua del directorio hasta que queden almacenados tantos backups como indique el argumento.

Según el procedimiento mencionado en el párrafo anterior, las operaciones de mantenimiento de backups se realizan mediante el identificador de la máquina. Pero si una máquina es eliminada o migrada se suprime de "/etc/qemu-server/", el fichero de configuración de dicha máquina, y por tanto el script de backup no puede conocer la existencia de esa máquina, dejando restos de backups antiguos en almacenamiento local. Para subsanar esto, se realiza una operación adicional en el script de backup consistente en comprobar los directorios que quedan dentro del directorio de backup y eliminar los pertenecientes a máquinas inexistentes. No obstante, en caso de que se haya realizado una migración, se debería de mantener el backup de esa máquina al menos una semana, para no tener una máquina sin puntos de restauración. · Es por esto por lo que se introduce un nuevo parámetro en el script (<DIAS>) que define el número de días que puede estar en disco un backup sin la máquina al que pertenece. Para esto, hay que comprobar si la diferencia de la fecha de modificación del backup con el momento actual es mayor o igual al parámetro especificado, en cuyo caso eliminará el backup. Con esto se soluciona el problema de dejar restos de backup que llenen el disco.

El modo de enlazar el backup local con el backup corporativo consiste en dejar en un directorio específico "/var/lib/vz/backup/" el backup que deseamos. El servicio de backup corporativo se encarga de acceder a ese directorio y copiar los ficheros que contenga. El directorio de backup corporativo debe contener exclusivamente los ficheros de la copia de seguridad actual de las máquinas virtuales de teletrabajo. De este modo el proceso consiste en eliminar la carpeta del backup corporativo antes de realizarlo, y volver a crearla con copias de los backups recién creados. Realizar esta operación implicaría tener dos backups por cada máquina virtual redundantes, de manera que se recurre al uso de enlaces para tener el fichero en dos directorios distintos sin requerir el almacenamiento de los dos.

```
Uso: ./vmbackup [-c <NUM> <DIAS> <MAQ_INICIO> <MAQ_FIN> | -r [<VMID> [<BACKUP>]]]

Maquinas disponibles a las que restaurar una copia de seguridad:
110
111
112
113
114
115
116
117
499
```

Ilustración 64 Script mostrando backups disponibles

El mismo script que se utiliza para realizar los backups es el utilizado para restaurarlos, si se utiliza la opción "–r" al ejecutar el comando. El proceso de restauración se encargará de leer el directorio de backups, donde se almacenan los backups de cada máquina en un directorio nombrado con el identificador de la máquina. Si además se le indica al script qué máquina se desea restaurar, lee el directorio correspondiente y muestra los backups almacenados para esa máquina, los que incluyen en su nombre la fecha y hora de creación del backup.

```
crisol@prox1:~$ /usr/local/bin/vmtools/vmbackup -r 111
Uso: ./vmbackup [-c <NUM> <DIAS> <MAQ_INICIO> <MAQ_FIN> | -r [<VMID> [<BACKUP>]]]

Copias de seguridad disponibles para la maquina 111:
vzdump-qemu-111-2011_12_07-21_30_22.tgz
vzdump-qemu-111-2011_12_11-21_30_17.tgz
crisol@prox1:~$ /usr/local/bin/vmtools/vmbackup -r 111 vzdump-qemu-111-2011_12_11-21_30_17.tgz
```

Ilustración 65 Restauración de máquina virtual con el script

En la última línea de *Ilustración 65 Restauración de máquina virtual con el script* aparece la sentencia con la que se restaurará el backup de una máquina virtual. El script, una vez localizado el fichero de backup a restaurar, lo hace apoyándose de la herramienta *qmrestore*, que se encarga de crear una máquina virtual a partir de un backup, siendo necesario previamente detenerla y eliminarla mediante el uso del comando *qm*. El código de este script está incluido en 10.3 Script de backup de máquinas virtuales.

El backup cruzado se realiza con un script diferente. El backup cruzado requiere del intercambio de datos entre dos nodos del clúster mediante el protocolo *ssh*. Una sesión normal de *ssh* requiere autenticar al usuario mediante su nombre y contraseña, al tratarse de un script, ese usuario y contraseña quedarían almacenados en texto en claro dentro del script, siendo susceptible de ser leída y suponiendo un importante problema de seguridad. Para evitar tener que introducir la contraseña en la conexión *ssh*, se configuran los nodos del clúster para que se identifiquen mediante clave pública.

En primer lugar se crea la clave del servidor mediante el comando:

```
ssh-keygen -b 4096 -t rsa
```

Lo que genera en el directorio "/home/usuario/.ssh/" la clave privada en el fichero "id_rsa" y la clave pública en "id_rsa.pub".

La clave pública generada se copia en el segundo servidor:

```
cd ~
scp .ssh/id_rsa.pub prox2:.ssh/prox1.pub
```

Una vez está copiada la clave se incluye en el llavero de claves de *ssh* del segundo servidor:

```
ssh prox2
cd .ssh
cat prox1.pub >> authorized_keys
rm prox1.pub
```

Tras repetir la operación en ambos servidores, se pueden realizar conexiones *ssh* y transmitir archivos mediante *scp* sin necesidad de introducir contraseñas.

El funcionamiento del script de backup cruzado consiste en acceder al directorio "/etc/qemu-server/" de la otra máquina del clúster y, mediante *scp,* copiar todos los ficheros de configuración de las máquinas virtuales en un directorio local "/etc/qemu-server/<nombre_maquina>/<fecha>", siendo <nombre_maquina> el nombre del servidor original donde estaban los ficheros de configuración y <fecha> el momento en el que se realizó la copia. Puesto que el script ofrecerá soporte para copias de seguridad de varios estados de las máquinas, es necesario que se encargue de eliminar las copias obsoletas de los ficheros de configuración, por ello el script recibe un único argumento en el que se le indica el número de días que se mantendrán las copias almacenadas. Tras realizar la copia, se comprueba la fecha de los directorios dentro de "/etc/qemu-server/<nombre_maquina>/" eliminando todo directorio cuya antigüedad en días sea mayor al indicado en el argumento. El código de este script está incluido en *10.2 Script de backup cruzado*.

Como se puede comprobar, no se tiene en cuenta en los scripts cuándo deben ser lanzados, ya que de esa tarea se encargará el proceso *crond*. Ambos scripts requieren la especificación de argumentos y opciones en la invocación para su funcionamiento, que habría que incluir en la configuración del *crond*. La modificación en un futuro de cualquiera de los parámetros, implica la modificación directamente en el fichero del *crond*. Por simplicidad se crea un tercer script que no recibe argumentos, sino que define como constantes los parámetros del backup, e invoca ambos scripts de backup consecutivamente. De esta manera, modificar cualquier parámetro del backup se

limita a ajustar la constante deseada dentro del script denominado "**backup**" y la configuración de *crond* se simplifica al tener únicamente que definir la ejecución de un único script sin argumentos. El script de backup está incluido en *10.4 Script de backup completo*.

Para programar el backup hay que editar en ambos servidores el fichero "crontab":

```
sudo vim /etc/crontab
```

Y añadir al fichero la siguiente línea:

```
# m h dom mon dow user command
00 21 * * 0,3 root  test -x /usr/sbin/anacron || (cd / && /usr/local/bin/vmtools/backup )
```

La línea indica que a las 21:00 de cualquier día del mes y cualquier mes, si es domingo o miércoles, ejecute el comando backup.

Para que la programación funcione deben estar localizados los tres scripts desarrollados dentro de la ruta "/usr/local/bin/vmtools/".

4.5.3. Plan de despliegue del servidor de monitorización

El servidor de monitorización constará únicamente de una máquina virtual alojada en un servidor de la Oficina de Software Libre. Teniendo que ser creada previamente una máquina virtual con las especificaciones detalladas en el apartado *4.3.1 Infraestructura Hardware*.

Para la creación de la máquina virtual se accede al servidor de Proxmox y, tras identificarse como usuario del mismo, se accede a la pantalla de máquinas que se puede ver en la *Ilustración 66 Pantalla de Máquinas Virtuales de Proxmox*.

Ilustración 66 Pantalla de Máquinas Virtuales de Proxmox

En esta pantalla se selecciona "Crear" y la interfaz muestra la pantalla que se muestra en Ilustración 67 Pantalla de creación de máquina en Proxmox. En la pantalla que se muestra es donde se definen los parámetros que definen la máquina virtual:

- **Tipo**: Por defecto está establecido en OpenVZ que es la opción más sencilla para virtualizar servidores Linux. Sin embargo, implica compartir el sistema operativo con la máquina anfitriona y se desea utilizar Debian en lugar de Proxmox, estableciéndose el tipo de virtualización en **KVM**.
- **Espacio del disco (GB)**: Se establece en **25**. El componente que más espacio de almacenamiento consumirá es la base de datos, sobre todo si se tiene en cuenta que el sistema de monitorización se va a extender a otros servicios que no son el de Teletrabajo y, por tanto, incrementar su tamaño. En lugar de estimar el espacio que necesitará para la base de datos, puesto que se tiene una máquina virtual, se prefiere asignar un tamaño inicial e ir incrementándolo en caso de ser necesario.
- **Nombre**: no tiene ningún efecto sobre la máquina, ya que solo sirve para identificarla más fácilmente por personas, en lugar de utilizar el identificador numérico. Se sigue la notación del resto de máquinas: el servicio que ofrecen y el sistema operativo instalado.
- **Memoria**: Se asignan **2048** MB (2 GB) siguiendo la misma filosofía que con el disco duro: ajustarla según las exigencias del sistema.

- **Iniciar al arranque**: **Activado**, lo que significa que cuando arranque el equipo anfitrión, arrancará automáticamente también esta máquina. De modo que ante cualquier problema que haga apagarse al equipo anfitrión, cuando sea arrancado se tendrá activo el servicio de monitorización ene l menor tiempo posible.
- **Tipo de S.O. invitado**: Puesto que Debian 6 es un sistema operativo moderno, cuenta con un núcleo de **Linux 2.6**.
- **CPU Sockets**: Aún no se pueden definir los núcleos que tendrá el procesador, a lo que se refiere este parámetro es al número de CPUs que tendrá la máquina, que se establece a **1**.

El resto de parámetros de la máquina virtual se dejan por defecto, y se selecciona la opción "créate", tras lo que quedará creada la máquina virtual.

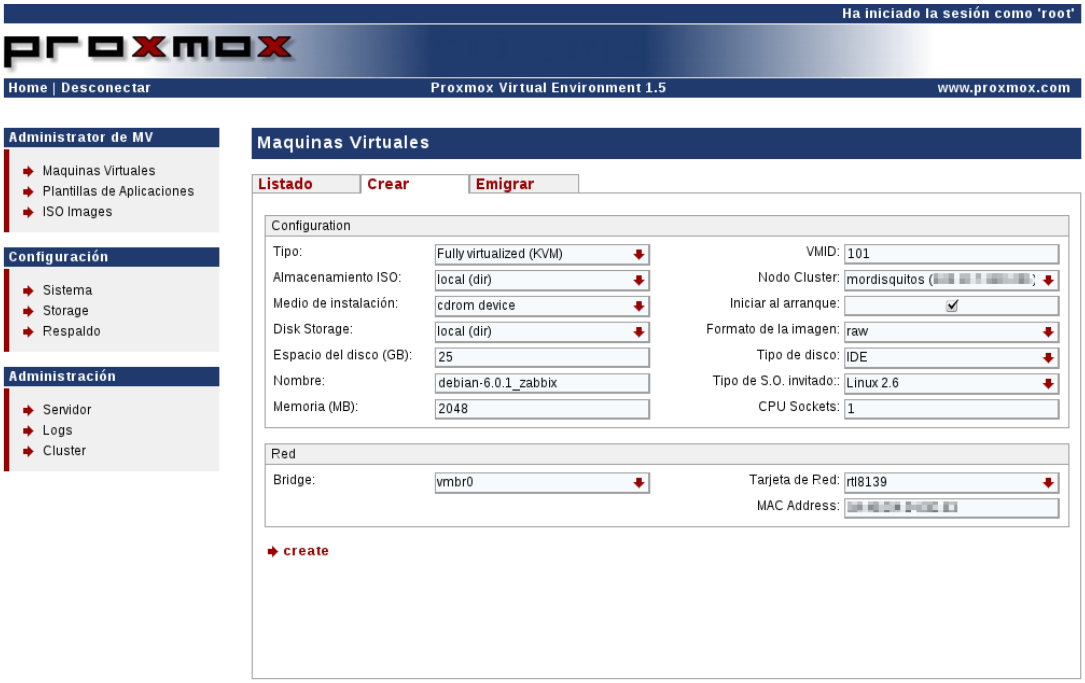

Ilustración 67 Pantalla de creación de máquina en Proxmox

Una vez creada la máquina virtual, aparecerá en la pantalla de máquinas virtuales de Proxmox (*Ilustración 66 Pantalla de Máquinas Virtuales de Proxmox*). Si se selecciona la máquina virtual creada, aparecerá la pantalla de la máquina virtual debían-6.0.1_zabbix, que permite realizar operaciones sobre la máquina como encenderla, apagarla, modificar su disco duro o memoria. Es aquí donde podemos definir el número de núcleos que tendrá la CPU de la máquina y que no se pudo definir durante su creación. Para ello se cambia el valor "1" que aparece en el cuadro

de texto “CPU Sockets” por 2, como se ve en *Ilustración 68 Pantalla de máquina virtual en Proxmox*.

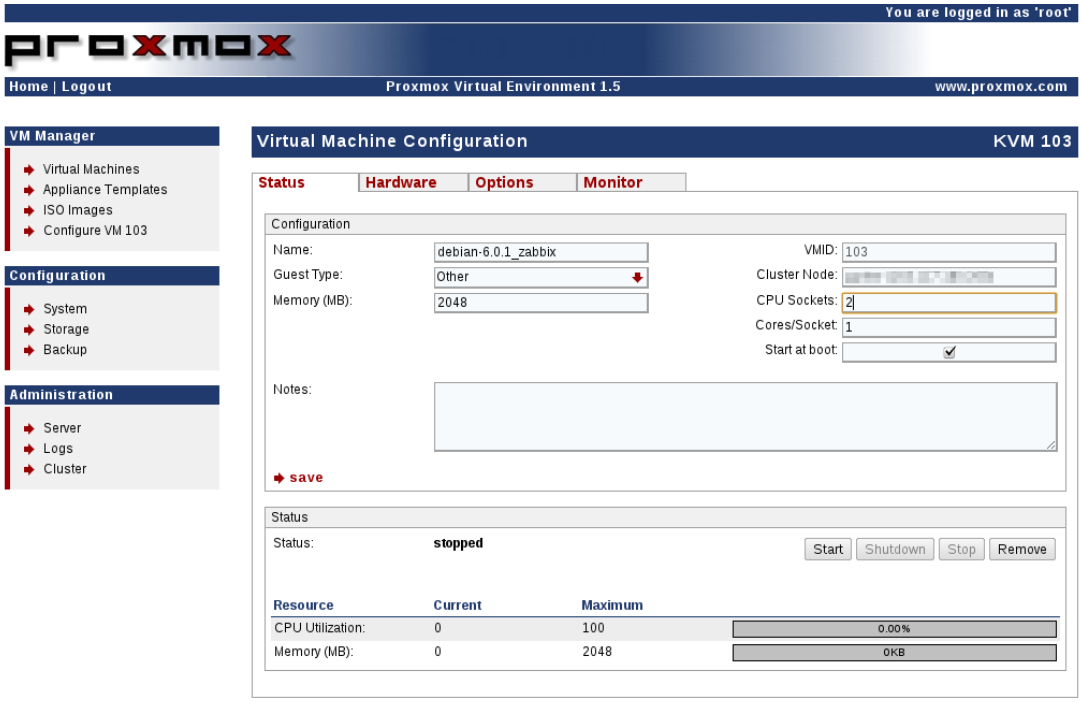

Ilustración 68 Pantalla de máquina virtual en Proxmox

Para realizar la instalación del sistema operativo hay que simular que se introduce un disco de instalación en la máquina virtual. Proxmox permite leer imágenes de disco en formato *iso* y simular que los datos que contiene es un disco óptico insertado en la unidad de CD-ROM de la máquina virtual.

Las imágenes *iso* tienen que estar en el directorio “/var/lib/vz/template/” del servidor Proxmox y, puesto que la web de Debian ofrece descargar su sistema operativo en este formato, se utilizan los siguientes comandos para descargar el sistema operativo para que lo encuentre Proxmox:

```
cd /var/lib/vz/template/iso/
sudo wget http://cdimage.debian.org/debian-cd/6.0.3/i386/iso-cd/debian-6.0.1-i386-CD-1.iso
```

Para introducir la imagen del disco del sistema operativo, se pincha sobre la pestaña “Hardware” de la máquina virtual en Proxmox y, a continuación, seleccionar la flecha roja que aparece bajo “CD-ROM device”, eso hará que aparezca la pantalla para editar la unidad de CD-ROM.

Cuando finalice la descarga, la imagen del disco de instalación de Debian debería de aparecer en el desplegable “CDROM”, se selecciona la imagen del disco de Debian

y, una vez se pinche sobre “apply”, ya estará el disco introducido en la máquina virtual.

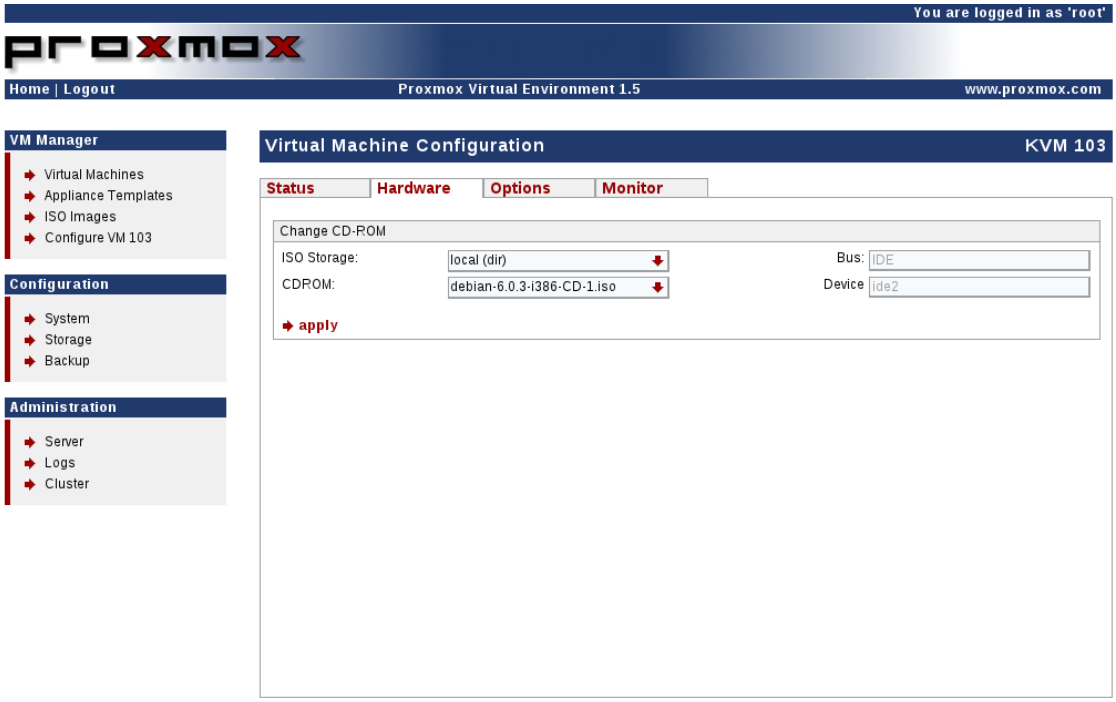

Ilustración 69 Pantalla de cambio de CDROM de una máquina virtual en Proxmox

Con la imagen del disco de Debian insertado en la unidad de discos de la máquina virtual, se enciende por primera vez la máquina virtual, la cuál se inicia arrancando el disco de Debian automáticamente y lanza el asistente de instalación. Por comodidad se decide realizar la instalación de forma gráfica, marcando la opción “Instalación gráfica” y pulsando la tecla “retorno”.

El asistente presenta una serie de preguntas comenzando por el idioma en el que queremos el sistema, a lo que se selecciona “español”, a continuación pregunta por el país en el que estamos, se selecciona “España” y por último pregunta por el teclado que tiene la máquina, a lo que se le indica también que “español”.

En este momento intenta configurar la red y su conexión a Internet automáticamente, pero al no disponer la red de un servicio de configuración automática, el instalador nos informa de que no se ha podido configurar la red porque no hay servidor de DHCP. Para configurar la red se selecciona la opción “Configurar red manualmente” y se pulsa “Continuar”. Esto iniciará un asistente de configuración de la red que pedirá que se introduzca la dirección IP de la máquina, la máscara de red de la máquina, la pasarela, direcciones de los servidores de nombres (DNS), el

nombre de la máquina (donde se nombra como zabbixosl) y el nombre de dominio, en el que se introduce el dominio de la Universidad: uc3m.es.

Configurada la red, comienza el proceso de creación de usuarios, en primer lugar crea el usuario "root" con todos los privilegios, para el que el asistente solicita una contraseña para él en el diálogo "Introducir clave de súper-usuario". A continuación pide que se cree un usuario, para el que primero solicita su nombre completo (nombre y apellidos), seguidamente solicita un nombre de usuario para el sistema y la contraseña con la que accederá ese usuario.

Tras preguntarnos por la zona horaria e indicarle que estamos en la "Península" comienza el proceso de particionamiento de discos. Cuando nos presenta la opción sobre como particionar, se escoge la opción "Manual" para evitar que Debian tome las decisiones sobre particionamiento. Se selecciona el disco que queremos particionar: "SCSI1 (0,0,0) (sda) – 26.8 GB ATA QEMU HARDDISK", y el asistente preguntará si se quiere crear una nueva tabla de particiones vacía para ese disco y se responde que "Sí". Se crea la tabla de particiones del disco y aparece en el menú "pri/lóg 26.8 GB ESPACIO LIBRE" se selecciona y pulsa "Continuar", en la siguiente ventana se pregunta cómo usar el espacio libre, se selecciona "Particionar de forma automática el espacio libre", una vez seleccionada la opción, se escoge "Separar particiones /home, /usr, /var y /tmp" que es la configuración recomendada para servidores.

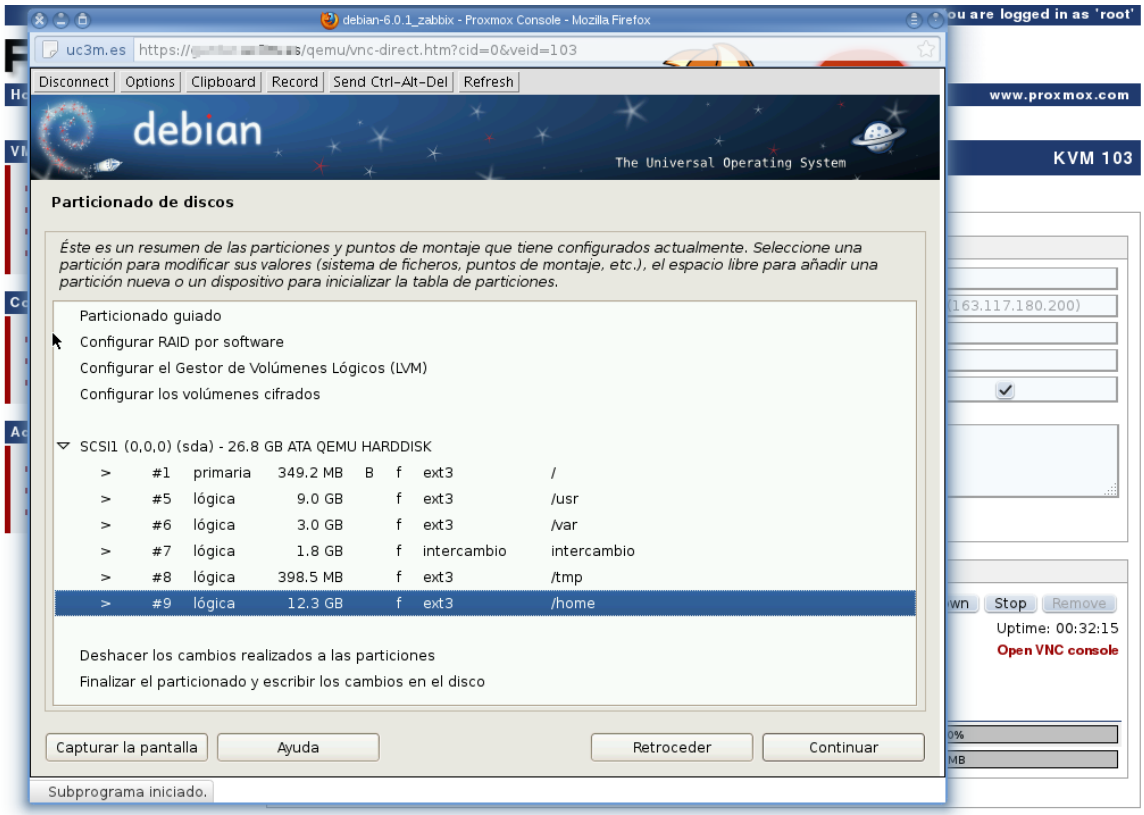

Ilustración 70 Particionamiento por defecto de Debian

Las particiones que genera Debian de forma automática aparecen en *Ilustración 70 Particionamiento por defecto de Debian*, este particionamiento no nos es conveniente porque, por ejemplo, dedica mucho espacio a la partición */home*, que apenas será usada y, sin embargo, tan solo 3 GB a la partición */var* donde se localizará la base de datos. Se quiere dedicar el máximo espacio posible a */var*, además, puesto que las consultas e inserciones serán muy frecuentes, se desea que la velocidad sea máxima, para lo que es recomendable el uso de una partición ext4 en lugar de la ext3.

Reducir el resto de particiones para dedicar más espacio a */var*, implica que el ya de por sí poco espacio dedicado a esas particiones, resulte ridículo o inexistente. Al no saber exactamente cuándo dedicar a cada una, la mejor opción es no realizarlas y estar todas localizadas en una partición raíz, en la que se distribuya el espacio acorde a las necesidades de cada carpeta. Por tanto se deja una partición "/" de 8,2 GB que es suficiente para la instalación del sistema operativo y el software de monitorización, y otra partición "/var" de 18 GB con el sistema de ficheros ext4. El resto de espacio no particionado se dedica a una partición extendida utilizada para intercambio.

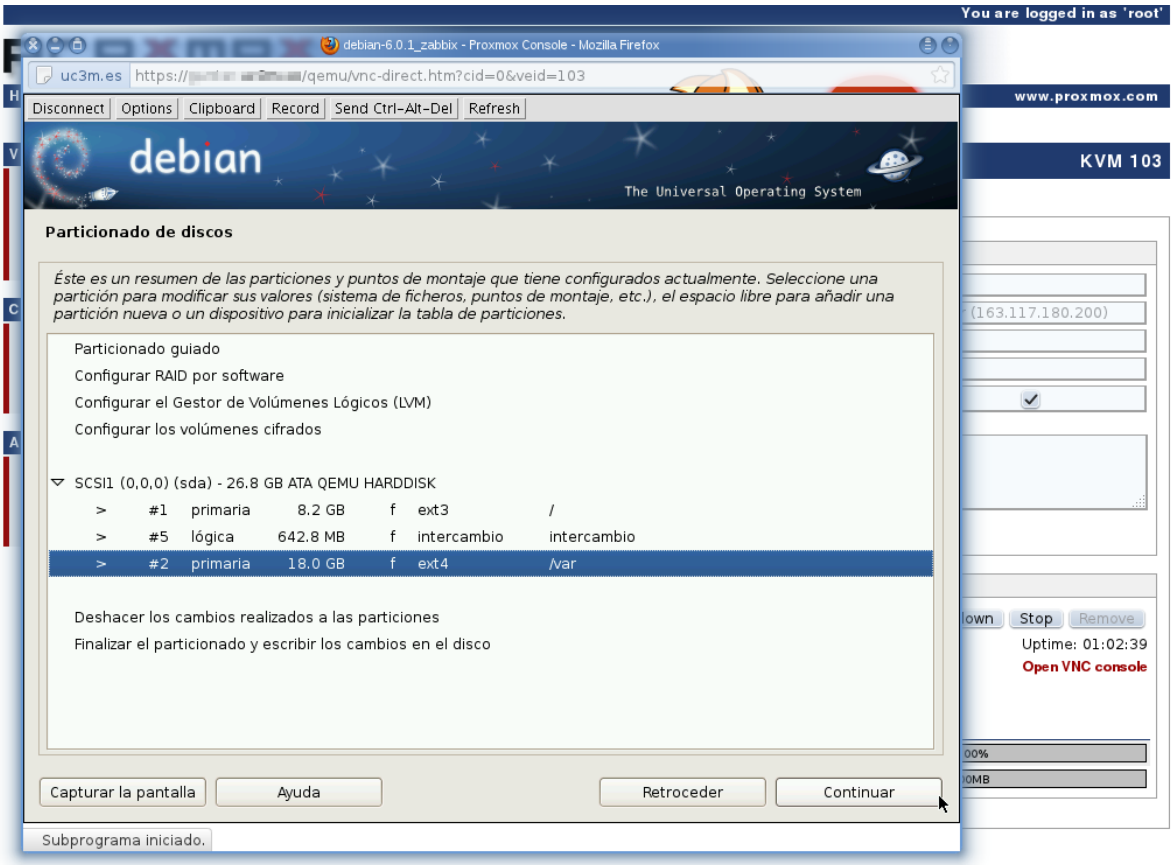

Ilustración 71 Partición personalizada

Para editar cada elemento de la partición basta con seleccionarlo y darle a continuar, tras lo que se muestra una pantalla con los datos de la partición, se selecciona por separado cada uno y se modifica, obteniendo el particionamiento deseado mostrado en *Ilustración 71 Partición personalizada*. Distribuido el espacio del disco, se selecciona "Finalizar el particionado y escribir los cambios en el disco" y se pulsa "Continuar", tras lo que pedirá una confirmación: "¿Desea guardar los cambios en los discos", se marca "Si" y se pulsa continuar para que el instalador comience a realizar y formatear las particiones.

Cuando termine de realizar las particiones del disco, comenzará automáticamente la instalación del sistema. Una vez se ha copiado el contenido del disco, el asistente pregunta si se desea continuar la instalación con otro disco, dejamos marcado "No" y pulsamos "Continuar", puesto que el software adicional que requiera la instalación se prefiere descargarlo directamente desde el repositorio de Debian a través de Internet una vez esté funcionando el sistema, motivo por el que se responde "Si" a pregunta "¿Desea utilizar una réplica de red?" que se refiere precisamente a esto. Para configurar la réplica se escoge la opción "España" para que nos de la lista de servidores más cercanos y, entre ellos, se elige "ftp.gul.uc3m.es" que, estando en la propia Universidad, nos proporcionará una velocidad de descarga mayor. Se deja en

blanco el cuadro de texto cuando pregunte por un proxy ya que no hay ninguno para acceder a Internet.

Estando el sistema base instalado aparece un nuevo menú en el que nos pregunta que software adicional se desea instalar. La instalación del software de monitorización se detalla en el siguiente punto, así que para finalizar la instalación se selecciona "Continuar", tras elegir únicamente "Servidor SSH" para poder acceder remotamente al equipo sin necesidad de tener que acceder a Proxmox. Cuando termina de instalar, actualizar y configurar el software nos pregunta si se desea instalar el cargador de arranque, a lo que se contesta que "Sí" para que pueda arrancar el sistema operativo. Se deja que configure el cargador "Grub" y al finalizar indicará que ha terminado la instalación.

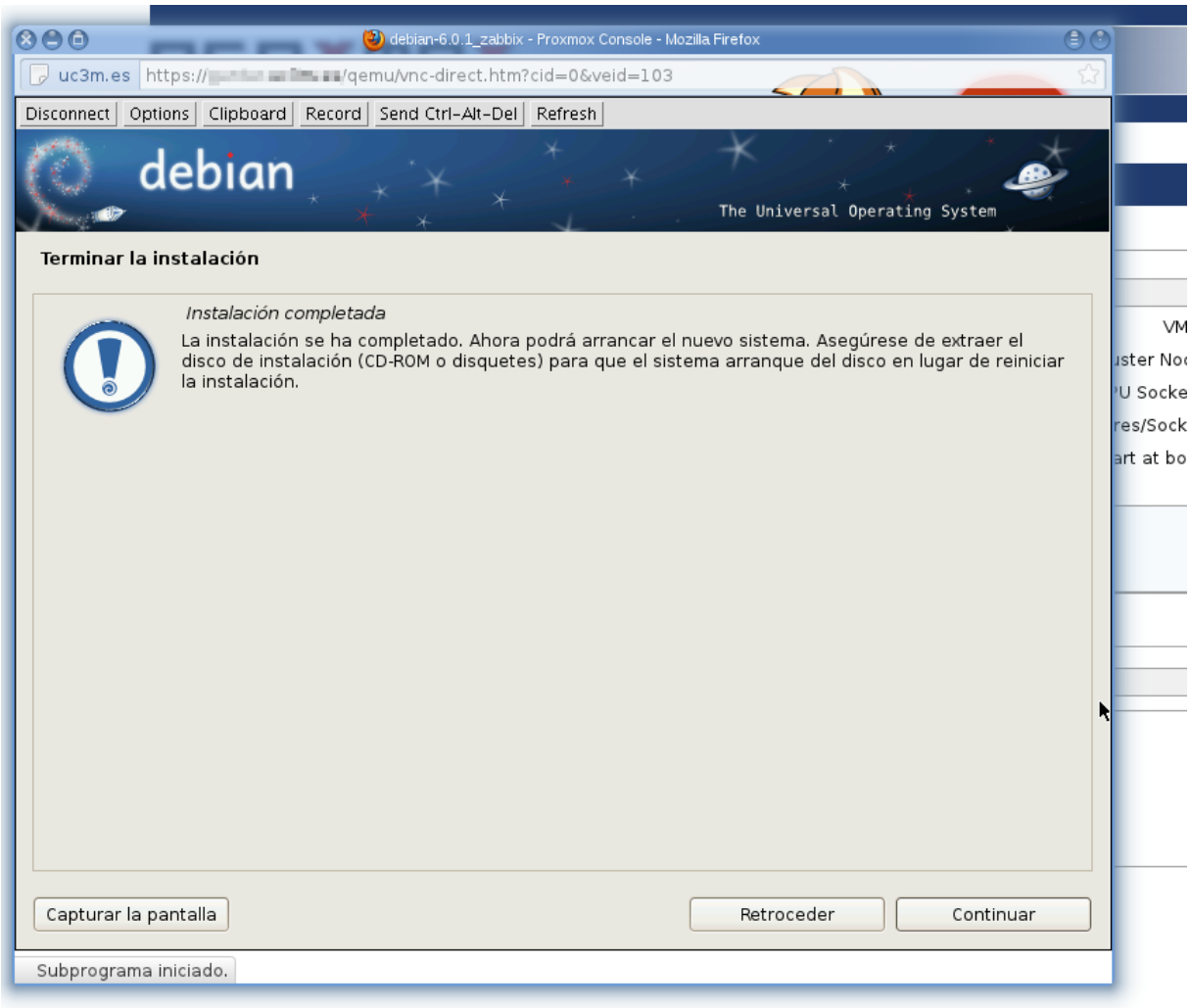

Ilustración 72 Instalación terminada de Debian

En cuanto se pulse sobre el botón "Continuar" reiniciará el equipo al estar al instalación completa. Para poder arrancar el sistema instalado hay que extraer

primero el disco de instalación, por lo que se accede a la pantalla “Hardware” de Proxmox, se pincha en la flecha roja bajo “CD-ROM drives” y se selecciona “Edit” (“Delete” implicaría eliminar la unidad entera, no sacar el disco). En el desplegable “CDROM” que se muestra, se selecciona “none (eject)” y se pulsa sobre “apply” para guardar los cambios. Se finaliza pulsando “Continuar” en el instalador de Debian.

4.5.4. Instalación del servicio de monitorización

Existen distintas maneras de instalar el servidor y la interfaz web de Zabbix. Por un lado tenemos las indicaciones del manual, en el que se detalla cómo realizar una instalación a partir del código fuente de Zabbix. Esta instalación requiere que se realice una instalación previa de la base de datos que se desee utilizar (MySQL, PostgreSQL, Fast...) el conector de la base de datos instalada con PHP y el propio PHP 5, en caso de realizar la instalación de la interfaz Web.

Por otro lado, existe una alternativa a la instalación más sencilla de instalar y robusta a problemas de compatibilidad e integración entre componentes, además de permitir actualizaciones automáticas de cada uno de los componentes implicados en la instalación y la corrección de fallos de software o vulnerabilidades, de una forma casi transparente al usuario. A cambio de estas ventajas hay que aceptar una desventaja, respecto a la opción de compilar el código fuente directamente, y es que raramente en los repositorios de la distribución en cuestión de GNU/Linux encontraremos la última versión del software, algo especialmente patente en la distribución escogida (Debian), en la que se prima la estabilidad de los paquetes por encima de la novedad de los mismos.

Para realizar la instalación de software y configuración del servidor, primero hay que permitir al usuario del sistema poder realizar operaciones de súper-usuario. Para ello se accede al sistema mediante *ssh* y se cambia de usuario a root y se ejecuta el comando *visudo*.

```
ssh zabbixosl.uc3m.es -l usuario
su
visudo
```

Se muestra un fichero en el que se debe añadir la siguiente línea:

```
usuario   ALL=(ALL) ALL
```

Se pulsa “Ctrl+X” para finalizar el programa, “S” para confirmar los cambios y “retorno” para mantener el fichero con el mismo nombre. Una vez finalizado, se cierra la sesión de root (comando *exit*), y se comprueba que el usuario puede utilizar el comando sudo.

Con los privilegios necesarios se comienza con la instalación del servidor de Zabbix mediante el comando:

```
sudo apt-get install zabbix-server-mysql
```

Saldrá una lista de paquetes adicionales que se deben instalar para el funcionamiento del servidor de Zabbix, entre ellos el gestor de bases de datos MySQL. Se pulsa la tecla “S”, para confirmar que se está de acuerdo con esas instalaciones adicionales, y comienza automáticamente el proceso de descarga e instalación.

Cuando estén instalados los componentes, aparecerá un asistente en la línea de comandos para configurar MySQL que preguntará qué contraseña se va a utilizar para acceder como administrador.

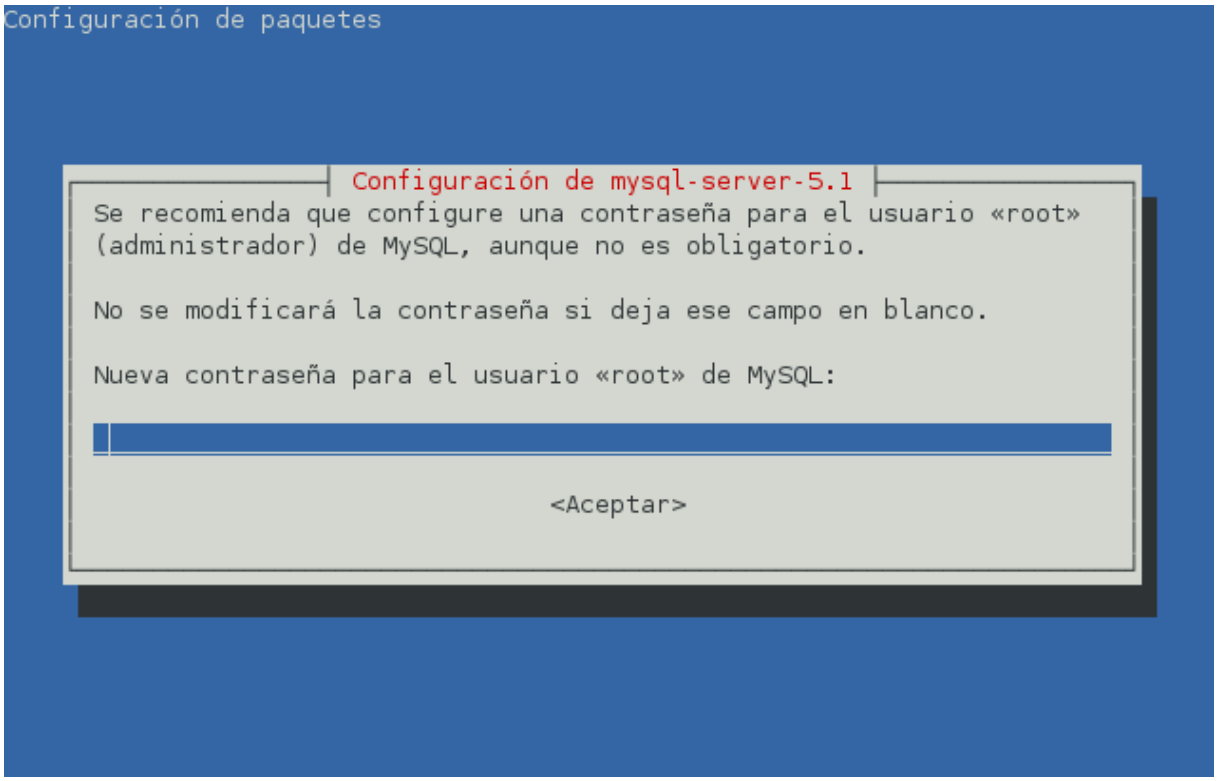

Ilustración 73 Asistente de configuración de MySQL

Definida la contraseña de MySQL, aparece un nuevo asistente para configurar el acceso por parte de zabbix-server-mysql a MySQL.

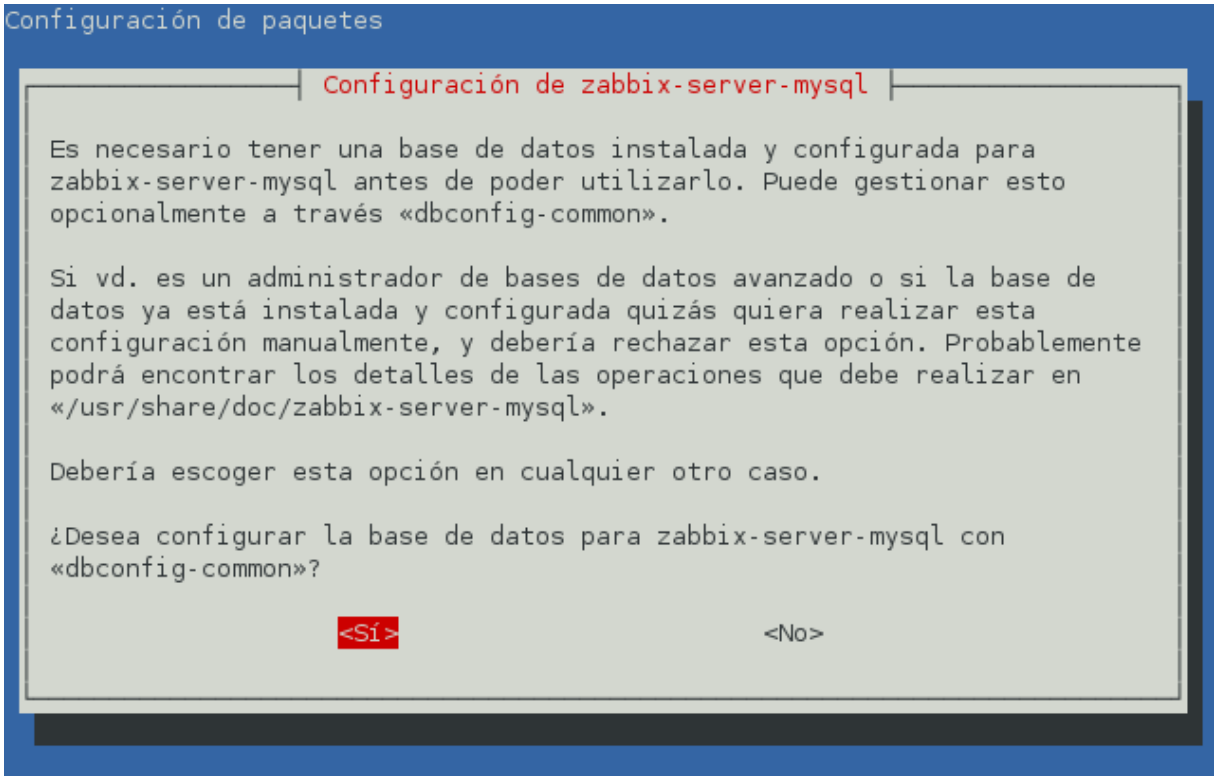

Ilustración 74 Asistente de configuración de la base de datos de Zabbix

El asistente preguntará si deseamos que se realice la configuración automática de la base de datos, a lo que se responde "Si" por simplicidad. Para poder configurarla, pedirá la contraseña de administración que se introdujo en el asistente anterior y, más adelante, pedirá una nueva contraseña para la nueva base de datos "zabbix". Por motivos de seguridad se introducen contraseñas distintas, tanto para el súper-usuario del sistema, como de administración de bases de datos y para la base de datos "zabbix". Cuando se le ha dado la contraseña, comienza a insertar automáticamente scripts en SQL con la definición de la base de datos. Una vez termine la inserción de scripts ya estará operativo el servidor de monitorización.

4.5.5. Configuración de la interfaz del sistema de monitorización

Aunque esté funcionando el sistema de monitorización, si no podemos indicar qué equipos y elementos queremos que sean monitorizados y, más importante, no existe un medio por el que transmitir al administrador las alertas o las mediciones recopiladas, no tiene ninguna utilidad el servicio. La interfaz es el componente que permite realizar todo esto y, además, ayuda a realizarlo de una forma gráfica y sencilla al administrador.

Para realizar la instalación de la interfaz, se accede al servidor de monitorización mediante *ssh* o a través de Proxmox, y se ejecuta el comando:

```
sudo apt-get install zabbix-frontend-php
```

La interfaz de Zabbix requiere para funcionar el entorno de ejecución de Apache, de modo que pedirá confirmación para instalar el servidor HTTP Apache 2 y PHP5, a lo que se contesta que si con la tecla “S”.

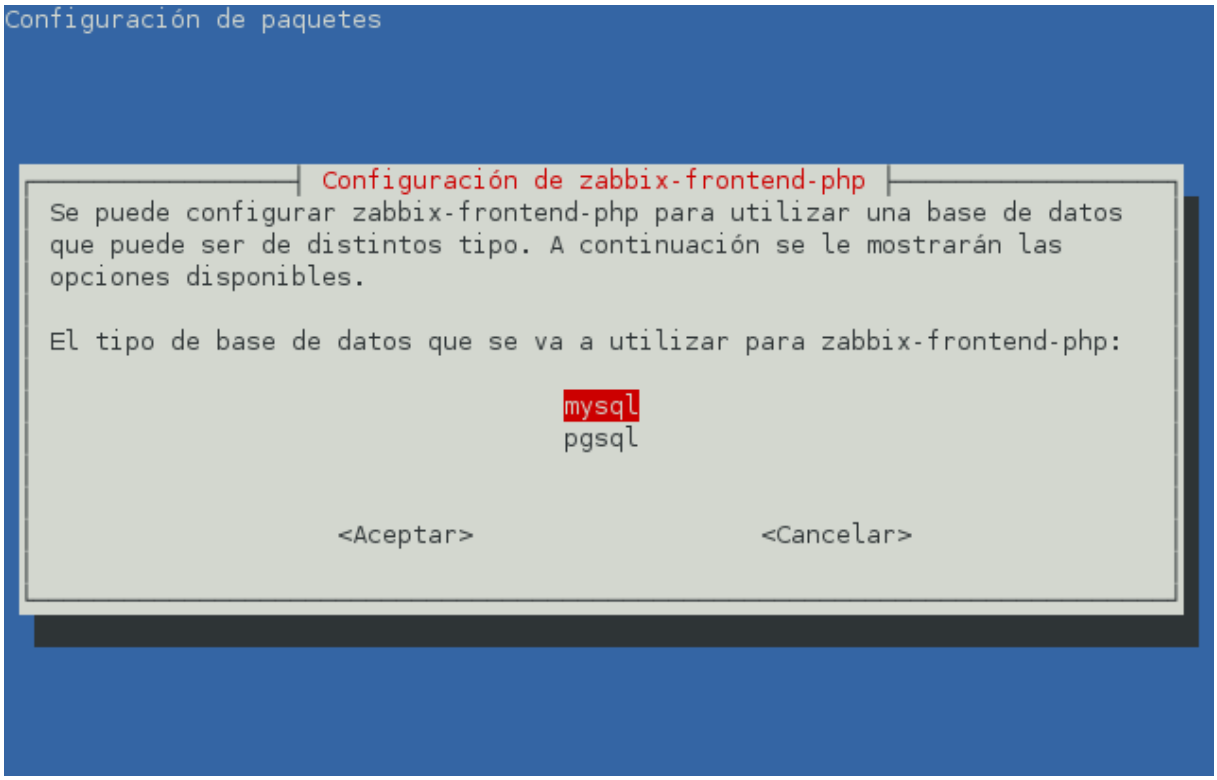

Ilustración 75 Asistente de configuracion del interfaz de Zabbix

Puesto que Zabbix puede utilizar distintos gestores de bases de datos diferentes, durante la instalación aparecerá un diálogo preguntando cuál es el que se está utilizando. Se selecciona “mysql” y preguntará por la contraseña de la base de datos. Se introduce la contraseña de la base de datos **Zabbix** y finalizará la instalación.

Antes de acceder a la interfaz se configurará el último componente que se gestiona desde línea de comandos: Exim4. Exim4 está ya instalado por defecto en Debian pero no está configurado. Para hacerlo se ejecuta el siguiente comando, que hará aparecer un asistente de configuración:

```
sudo dpkg-reconfigure exim4-config
```

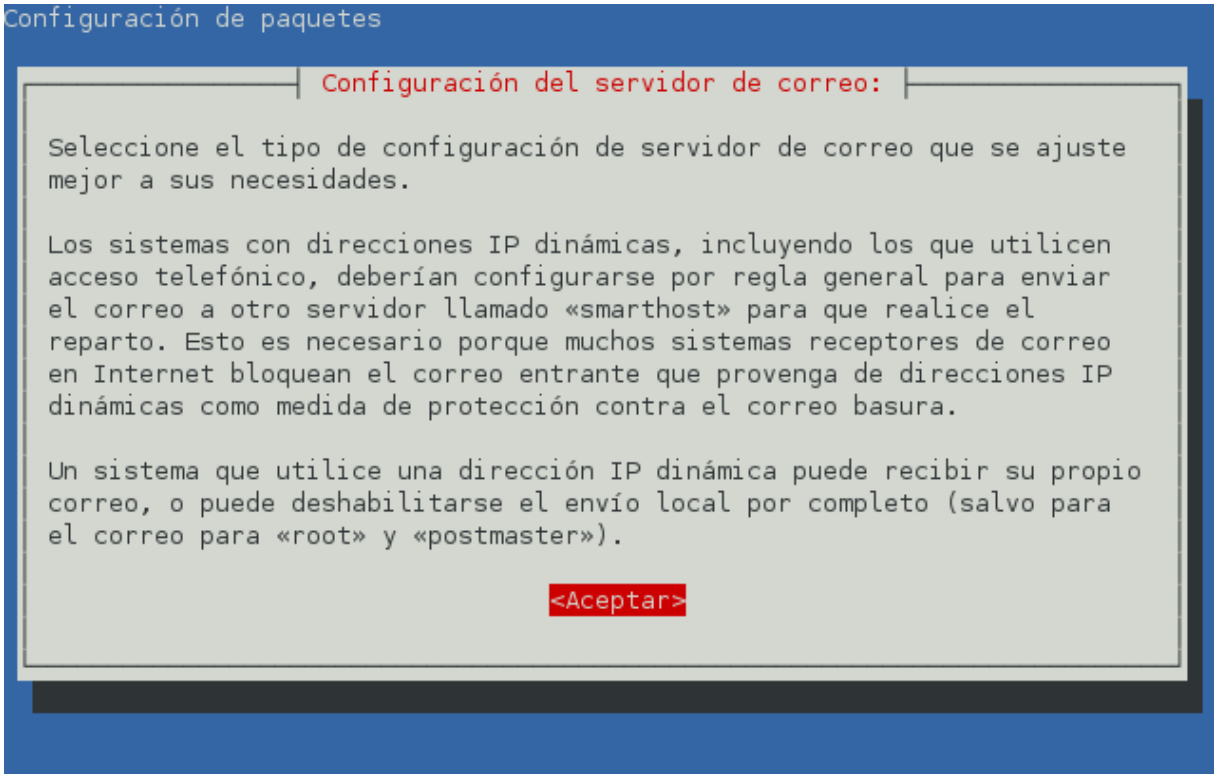

Ilustración 76 Asistente de configuración de Exim4

Tras aceptar el mensaje de la pantalla inicial de configuración, preguntará el origen y el destino del correo, y la forma de hacerlo. Se responde marcando la primera opción: "Internet site; el correo se envía y recibe directamente usando SMTP". A continuación preguntará sobre el nombre del servidor en el que se localiza la estafeta de correo, se introduce **zabbixosl.uc3m.es**. Puesto que la estafeta únicamente se encargará de recibir mensajes locales, provenientes de Zabbix, y enviarlos a través de internet, cuando el asistente pregunte cuáles son los servidores autorizados para recibir mensajes, se introduce la dirección local **127.0.0.1**. En el siguiente paso preguntará qué dominio debe interpretar como gestionadopor el servidor, este dominio es el nombre de la propia máquina **zabbixosl.uc3m.es**. Tras esto preguntará cuáles son los dominios de destino a los que debe mandar los mensajes, se introduce **uc3m.es,** ya que los administradores de monitorización recibirán las notificaciones en sus cuentas de correo de la Universidad. La siguiente opción sobre "smarthosts" se ignora, dejando el campo en blanco. El servidor tiene una conexión persistente Ethernet a la red, motivo por el que se responde "No" cuando nos pregunte si se deben limitar las consultas DNS por utilizar una conexión "dialup". Configuradas las funciones de servidor de correo, pasa a configurar la recepción de correos, preguntando el método que se desea utilizar para ello, se escoge: "formato mbox en «/var/mail»". Por último, preguntará como deseamos almacenar la configuración, si con un fichero grande como se hacía antiguamente, o si deseamos utilizar el nuevo formato de varios ficheros pequeños. Se responde que "No" a usar ficheros pequeños

ya que, al no ser expertos en configuración de Exim, resulta más difícil encontrar un parámetro determinado si no se sabe en qué fichero está.

Para acceder a la interfaz de Zabbix, se abre cualquier navegador web, y se introduce la dirección zabbixosl.uc3m.es/zabbix.

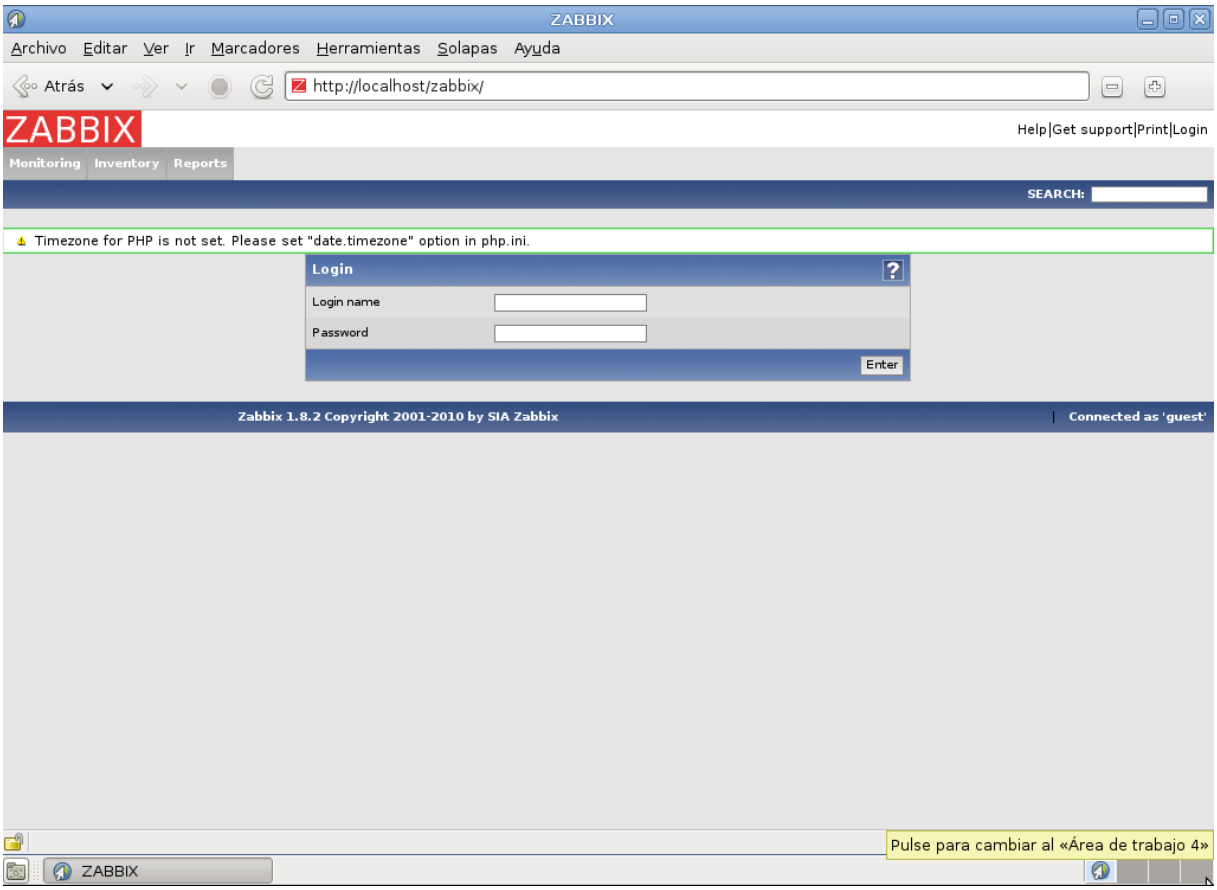

Ilustración 77 Pantalla de inicio de sesión de Zabbix sin configurar

En la *Ilustración 77 Pantalla de inicio de sesión de Zabbix sin configurar* aparece la pantalla para introducir el nombre y usuario para acceder al servidor de monitorización. Se puede observar que justo encima del cuadro de "Login" aparece un mensaje de advertencia diciendo que no está configurada la zona horaria en PHP. Se accede introduciendo el usuario por defecto, que es "admin" y la contraseña por defecto "zabbix".

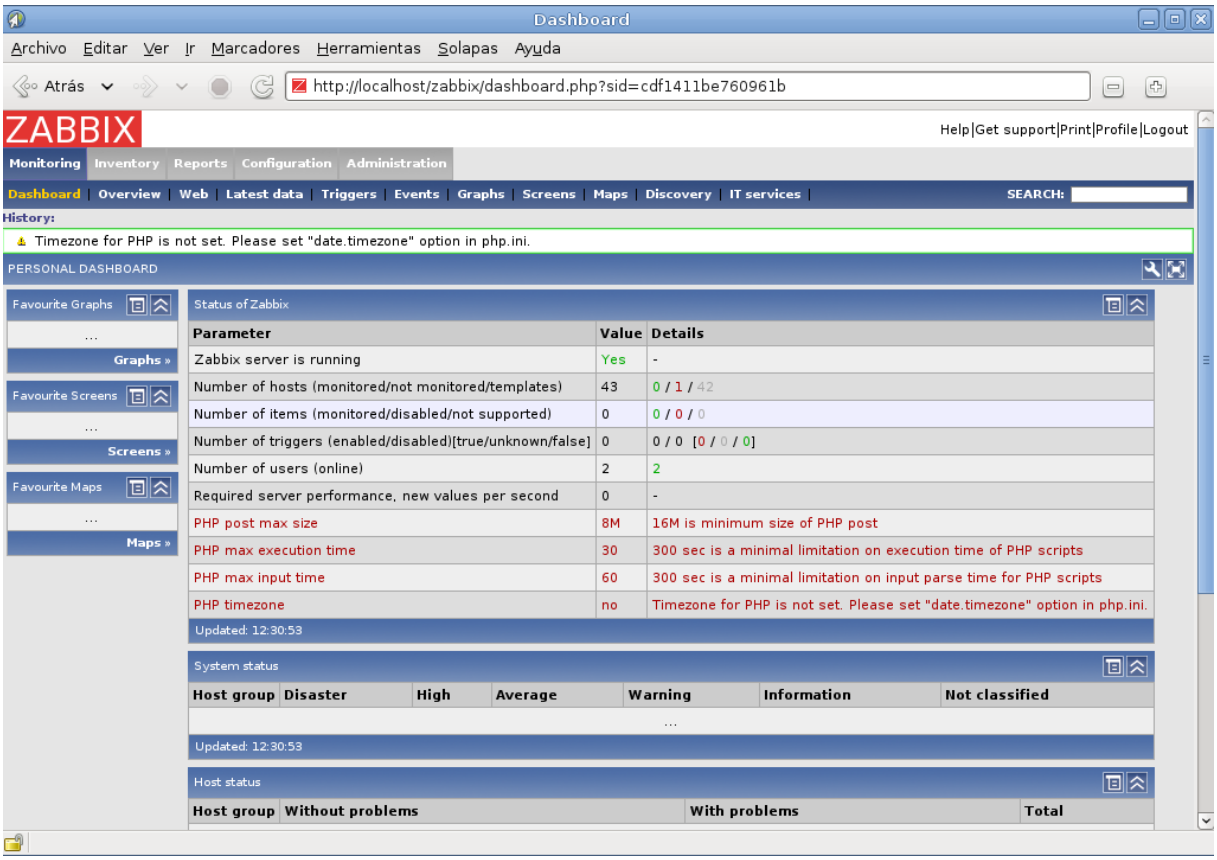

Ilustración 78 Pantalla Dashboad de Zabbix sin configurar

Una vez dentro del sistema, se ve como informa de cuatro problemas nuevos aparte del de la zona horaria. Estos problemas derivan en que a pesar de que la instalación de Zabbix configuró automáticamente la base de datos, no hace lo mismo con PHP, por lo que la configuración de PHP hay que realizarla a mano.

Para el correcto funcionamiento de la interfaz web de Zabbix, es necesario editar el fichero de configuración de PHP modificando algunos parámetros según nos indica el manual de Zabbix:

Parámetro	Valor mínimo
PHP Memory limit	128M
PHP post max size	16M
PHP max execution time	300
PHP max input time	300

Ilustración 79 Parametros a configurar de PHP

Los parámetros se establecen modificando el fichero php.ini que se encuentra en la ruta "/etc/php5/apache2/php.ini", se abre el fichero con *vim* y se buscan las siguientes líneas:

```
memory_limit = 128M

post_max_size = 8M

max_execution_time = 30

max_input_time = 60
```

Y establecer los nuevos valores reemplazándolas por:

```
memory_limit = 128M

post_max_size = 16M

max_execution_time = 300

max_input_time = 300
```

Adicionalmente hay que establecer un valor para la propiedad "date.timezone", que por defecto no está establecido en el fichero, dándole uno de los valores válidos definidos en el manual de PHP (The PHP Group, 2011), que en este caso es "Europe/Madrid".

Originariamente en el fichero php.ini, se puede encontrar la siguiente línea comentada:

```
;date.timezone =
```

Se elimina el comentario y se añade el valor seleccionado del manual de PHP.

```
date.timezone = "Europe/Madrid"
```

Una vez realizadas todas las modificaciones, hay que reiniciar el servicio apache para que se apliquen los cambios, para lo que únicamente hay que ejecutar el comando:

```
sudo /etc/init.d/apache2 restart
```

Finalmente se accede al sistema de monitorización a través de la interfaz web para comprobar que funciona correctamente, en caso de que se pueda acceder y no se dé ningún tipo de alerta, significa que la interfaz web funciona correctamente.

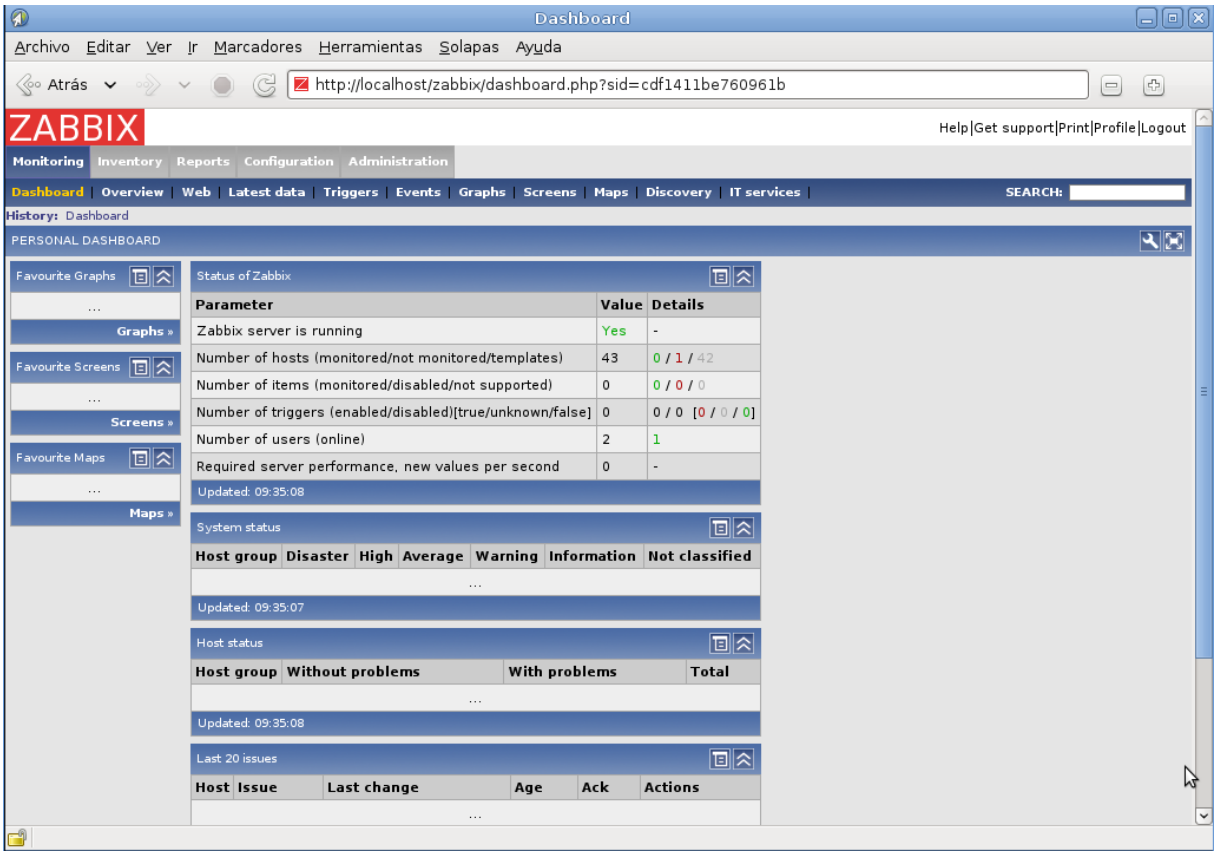

Ilustración 80 Pantalla Dashboard de Zabbix con PHP configurado

4.5.6. Configuración de agentes de monitorización

La instalación de los agentes de Zabbix en los equipos monitorizados sigue tres procedimientos, en función del sistema operativo de la máquina que se quiere monitorizar y, en caso de ser GNU/Linux, la versión más moderna del agente que está disponible en repositorios.

La instalación del agente en los equipos GNU/Linux, como son los servidores de Teletrabajo, se hace descargando e instalando el paquete zabbix-agent con el gestor de paquetes. Para el caso de los servidores de Teletrabajo, puesto que usan sistemas operativos basadas en Debian se instala con el siguiente comando:

```
sudo apt-get install zabbix-agent
```

La instalación crea un directorio en "/etc/zabbix/" que contiene dos ficheros de configuración "zabbix_agent.conf" y "zabbix_agentd.conf". El primer fichero sirve para configurar la versión *inetd* del agente, por lo que no será utilizado. El segundo es el que se debe editar ya que es el fichero de configuración del demonio de Zabbix que

es el que se ejecutará en todos los equipos. Abrimos el fichero con un editor de texto como *Vim* y se buscan los siguientes parámetros:

```
Server=

Hostname=
```

Dependiendo de la versión del agente de Zabbix instalado puede que los siguientes parámetros no existan, en cuyo caso habrá que definirlos.

```
;EnableRemoteCommands=

;AllowRoot=
```

Se completan o sustituyen las líneas del fichero para que tengan los siguientes valores:

```
Server=zabbixosl.uc3m.es

Hostname=<Nombre_maquina>

EnableRemoteCommands=1

AllowRoot=1
```

El valor de la línea "Server" puede ser dado tanto como dirección IP o nombre DNS. En general, para los equipos que se tiene acceso, se introduce la dirección IP para ahorrar las operaciones de resolución de nombre. En cambio, los equipos a los que no se tiene acceso se pone el nombre DNS para así, en caso de que el servidor de monitorización cambie, no haga falta modificar esos agentes.

Donde se hace referencia a <Nombre_maquina> hay que poner el nombre con el que queremos que se identifique este equipo en el servidor de monitorización, es importante que el nombre sea exactamente igual en ambos lados, si no surgirán algunos fallos en el log.

AllowRoot y EnableRemoteCommands se establecen a 1 para indicar que están activos. AllowRoot hace que zabbix-agentd sea lanzado como un proceso de root, poseyendo todos los privilegios de ejecución, de esta forma permite ejecutar remotamente comandos que requieran de estos privilegios, además, el segundo parámetro es el que permite realizar precisamente esto, permitiendo al servidor de Zabbix ejecutar comandos en el equipo monitorizado. Estas dos propiedades son

necesarias para el funcionamiento del script de firewall, que puede consultarse en 10.1 *Script de verificación de estado del firewall*.

La instalación del script de firewall se realiza desde cada máquina ejecutando el siguiente comando:

```
   sudo scp usuario@zabbixosl.uc3m.es:/usr/local/bin/firewall
/usr/local/bin/firewall
```

Siendo "usuario" cualquier administrador del servidor de Zabbix con acceso a ese directorio.

Los agentes de Zabbix anteriores a la versión 1.8 es necesario actualizarlos, pues en esta versión ha cambiado la forma en la que se mide el tamaño de datos. Todos los agentes con versiones de agente 1.8 miden este valor en bytes, en cambio los anteriores pueden hacerlo tanto en bytes como en kilobytes, dependiendo de en qué unidad haga la medición la plataforma en sí. Este hecho provocaba incoherencias en las mediciones y estadísticas extraídas entre distintas máquinas, en especial con los servidores de Teletrabajo. En vez de subsanar el problema a mano, realizando ajustes indicando como recalcular cada *item* implicado, se opta por actualizar la versión del agente en los equipos afectados. Para ello se descarga de la página web de Zabbix un paquete comprimido con los ejecutables binarios del agente de Zabbix compilado y se sustituyen los viejos. En el directorio "/usr/bin/" se copian los ficheros "zabbix_agent" y "zabbix_sender" y el fichero "zabbix_agentd" en "/usr/sbin/".

El hecho de realizar la actualización de una forma tan manual puede llegar a provocar problemas, en caso de que surgiesen actualizaciones para el paquete zabbix_agent en los repositorios de los sistemas implicados. Debido a la antigüedad de los paquetes, se ha asumido que ya no van ser actualizados, y en caso de ser así, basta con no autorizar esta actualización en concreto. Respecto a la alternativa de actualizar compilando el código fuente, ha tenido que ser descartada por los problemas que presentaba, no sólo las dependencias de bibliotecas para la compilación, sino problemas que surgían en la instalación de los scripts de inicio.

Tras modificar los ficheros de configuración del agente y actualizar los agentes, es necesario reiniciar el servicio de monitorización:

```
   sudo /etc/init.d/zabbix_agentd restart
```

Para la instalación del agente en Windows XP, se descarga desde la página web de Zabbix el agente para Windows de 32 y 64 bits, cuando finalice la descarga se tendrá un archivo comprimido. Al descomprimir el archivo aparecen dos directorios "win32" y "win64". Puesto que las máquinas virtuales de Teletrabajo son de 32 bits, se copia el directorio a "C:\Archivos de programa\Zabbix\" (no es necesario que sea

esta ubicación, pero se introduce ahí por organización de la máquina), y se elimina el directorio win64.

Dentro del directorio "C:\Archivos de programa\Zabbix\win32" se crea un fichero de texto plano con el nombre "zabbix_agentd.conf" con el siguiente contenido:

```
Server=zabbixosl.uc3m.es
Hostname=<Nombre_maquina>
```

En <Nombre_maquina> ponemos el nombre que se forma utilizando el prefijo "tt-" delante del nombre de usuario del teletrabajador.

Con el fichero creado abrimos una consola de Windows y se introducen los siguientes comandos para instalar y configurar el agente:

```
cd "C:\Archivos de programa\Zabbix\win32"
zabbix_agentd.exe --config "C:\Archivos de programa\Zabbix\win32\zabbix_agentd.conf" --install
```

Ahora hay que iniciar el agente y asegurarse de que se inicia automáticamente. Se pincha sobre el menú "Inicio" y en el menú contextual de "Mi PC" se selecciona "administrar", a continuación, en la lista que se presenta a la izquierda, se despliega el elemento "Servicios y Aplicaciones" y se selecciona "Servicios".

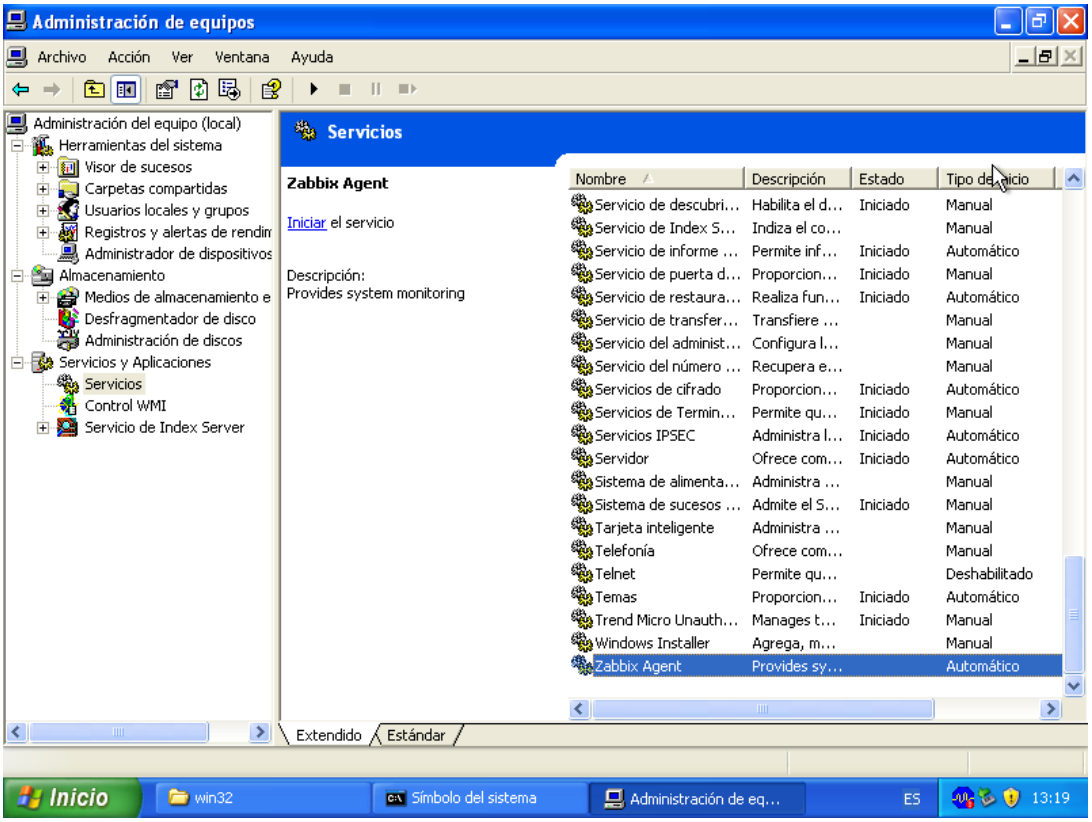

Ilustración 81 Pantalla de servicios en Windows XP

En la lista de servicios a la derecha buscamos el servicio llamado “Zabbix Agent” y, cuando se localice, se selecciona la opción “Propiedades” de su menú contextual. Se mostrará la pantalla de la *Ilustración 82 Pantalla de propiedades del servicio Zabbix Agent*, donde nos aseguramos de seleccionar en el desplegable “Tipo de inicio” el valor “Automático”. Tras pinchar sobre el botón “Iniciar” de la pantalla de propiedades de Zabbix Agent comenzará la ejecución del agente, el cual intentará establecer una conexión con el servidor de Zabbix.

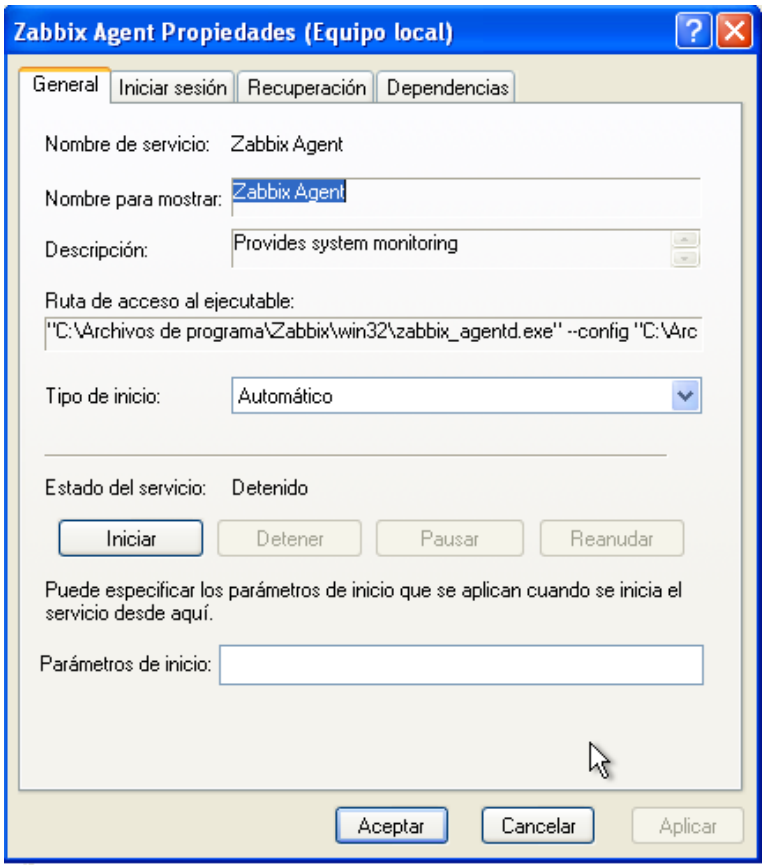

Ilustración 82 Pantalla de propiedades del servicio Zabbix Agent

4.5.7. Configuración del servicio de monitorización

Puesto que el sistema de monitorización seleccionado es Zabbix, a partir de este punto se utilizarán los términos que utiliza Zabbix para referirse a los conceptos sobre monitorización, según la siguiente correspondencia:

- “Elemento monitorizado” = “Item”

- "Alerta" = "Disparador" = "Trigger"
- "Equipo monitorizado" = "Host"

4.5.7.1. Definición de grupos

Estando los agentes instalados en los equipos se pude comenzar a configurar el servicio de monitorización. En primer lugar, como medida de organización se crean los grupos "OSL" y "Teletrabajo". El grupo OSL clasificará todas las plantillas de equipos que sean desarrolladas por la Oficina de Software Libre y los equipos monitorizados pertenecientes o administrados por este grupo, lo que incluye los dos servidores de Teletrabajo, el servidor de monitorización y el servidor de virtualización que lo hospeda. El grupo Teletrabajo, por su parte, contiene únicamente las máquinas virtuales del servicio de Teletrabajo.

Para definir un grupo se accede a la pestaña "Configuration" de la interfaz de Zabbix, y se selecciona el menú "Host groups". A continuación se pincha sobre el botón "Create group" y se presenta una pantalla en la que definir el nombre del grupo y seleccionar los equipos y plantillas que pertenecerán a ese grupo. En el cuadro de texto etiquetado como "Group name" se escribe el nombre del grupo (OSL o Teletrabajo) y se pincha sobre el botón "Save", tras lo que quedará definido el grupo.

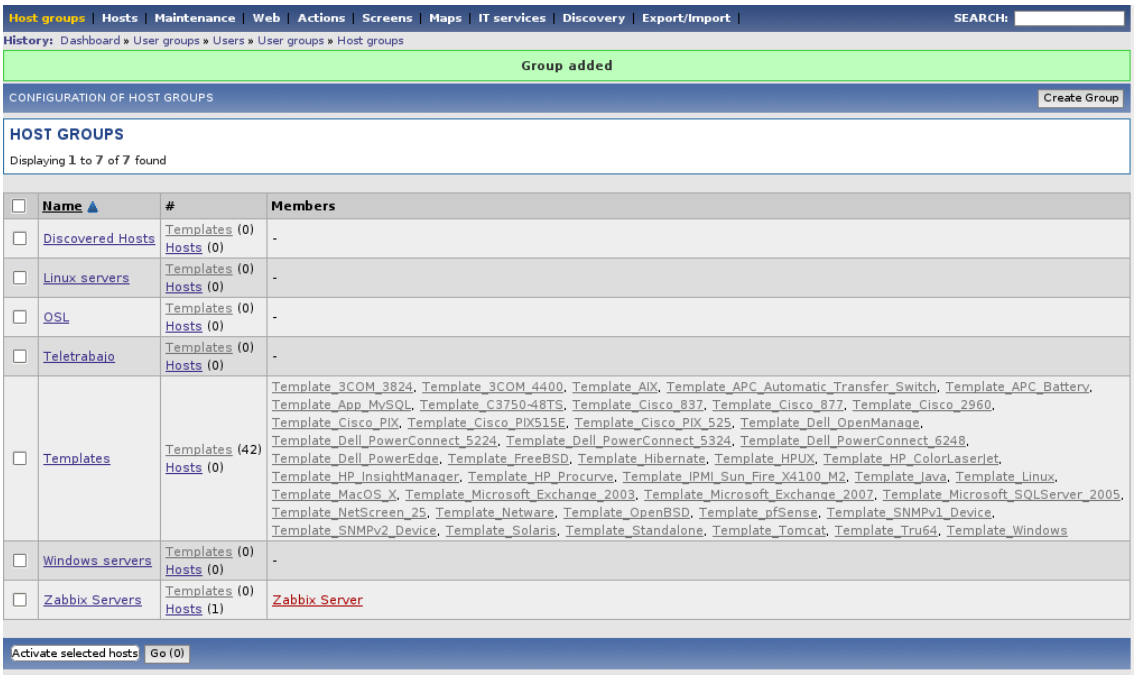

Ilustración 83 Pantalla de definición de grupos de Zabbix

4.5.7.2. Definición de usuarios

Para acceder al servicio de monitorización se accede con el usuario y contraseña por defecto, de modo que el primer paso es eliminar ese usuario y crear nuestros usuarios. Se ha decidido utilizar dos tipos de usuarios, un usuario administrador de la Oficina de Software Libre, que tenga acceso a modificar cualquier elemento de configuración del servicio de monitorización y además tenga la visión de todos los equipos monitorizados. Respecto al segundo usuario, se denominará "teletrabajo", y no poseerá ninguna capacidad de administración, sin embargo puede ver todos los datos, gráficas, alertas de las máquinas virtuales del servicio de Teletrabajo, además de recibir las notificaciones referentes a esas máquinas.

Para realizar esto, se accede a la interfaz de Zabbix, se selecciona la pestaña "Administration" y a continuación el menú "Users". Aparecerá una tabla con los roles y usuarios que pertenecen a esos grupos, en la columna "Members" se selecciona el usuario Admin.

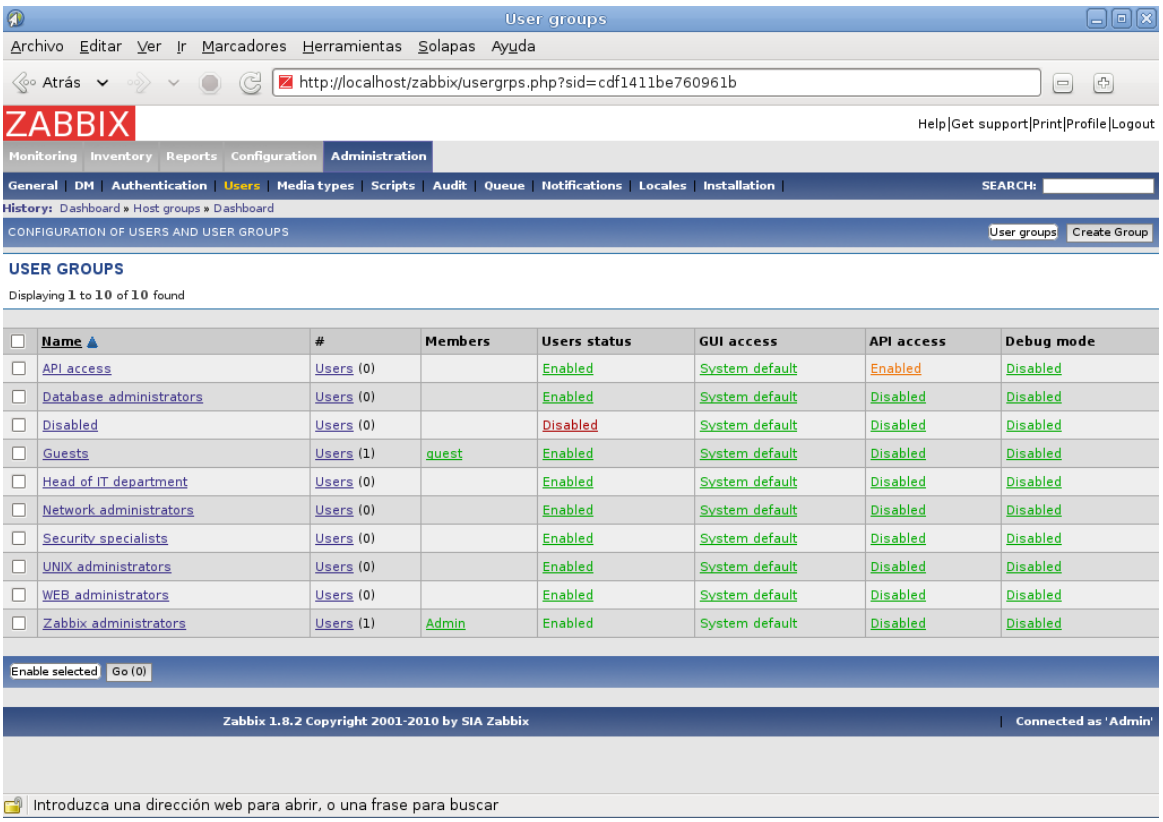

Ilustración 84 Pantalla de administración de usuarios de Zabbix

Se nos presentará la pantalla del usuario Admin, en el cuadro de texto "Alias" reemplazamos "Admin" por el nombre de nuestro usuario administrador y pinchamos sobre "Change password" para que aparezcan dos cuadros de textos en los que se introduce la nueva contraseña y la confirmación de la misma. Se nos presenta la opción de cambiar el idioma de la interfaz a castellano pero, puesto que la traducción nos presenta más confusiones que ayuda, se deja en inglés. En la parte inferior de la pantalla aparece una etiqueta "Media" y un botón "Add", pinchamos ese botón para que acceder a otra pantalla en la que se permite introducir medios de comunicación con el usuario, de modo que cuando surja una notificación, ésta será enviada al usuario a través de esos medios de comunicación. En el desplegable del medio se selecciona Email y en el cuadro de texto junto a la etiqueta "Send to" se escribe la dirección de correo electrónico del grupo OSL, se pincha sobre "Add" para añadir la dirección, y una vez de vuelta a la pantalla de definición de usuario, se pincha sobre "Save" para que quede definido el usuario.

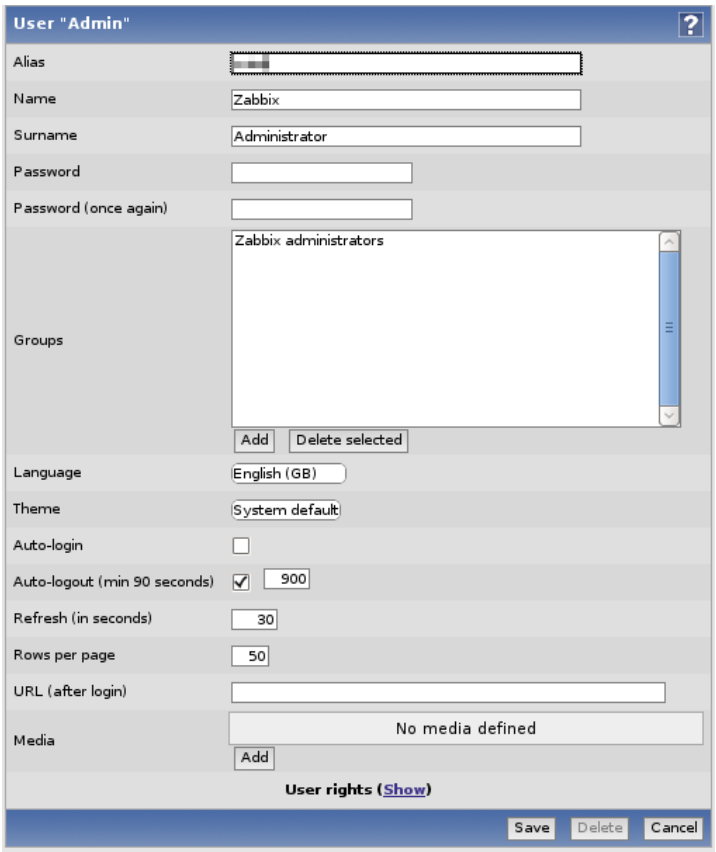

Ilustración 85 Pantalla de definición de usuario en Zabbix

El proceso para definir al usuario "teletrabajo" es análogo, pero antes hay que definir un rol que describa cuáles son las competencias de este usuario. La definición del rol se hace desde la pantalla de administración de usuarios (*Ilustración 84 Pantalla de administración de usuarios de Zabbix*), en esta

pantalla se pincha sobre “Create group”. En la pantalla que aparece de creación de grupo se establece su nombre (Teletrabajo) dentro del cuadro de texto etiquetado como “Group name”. En la parte inferior aparecen tres listas etiquetadas como “Rights”, en la primera de ellas, denominada “Read-write”, se pincha sobre “Add” y se mostrará una lista con los grupos del sistema, se selecciona Teletrabajo y a continuación “Select”.

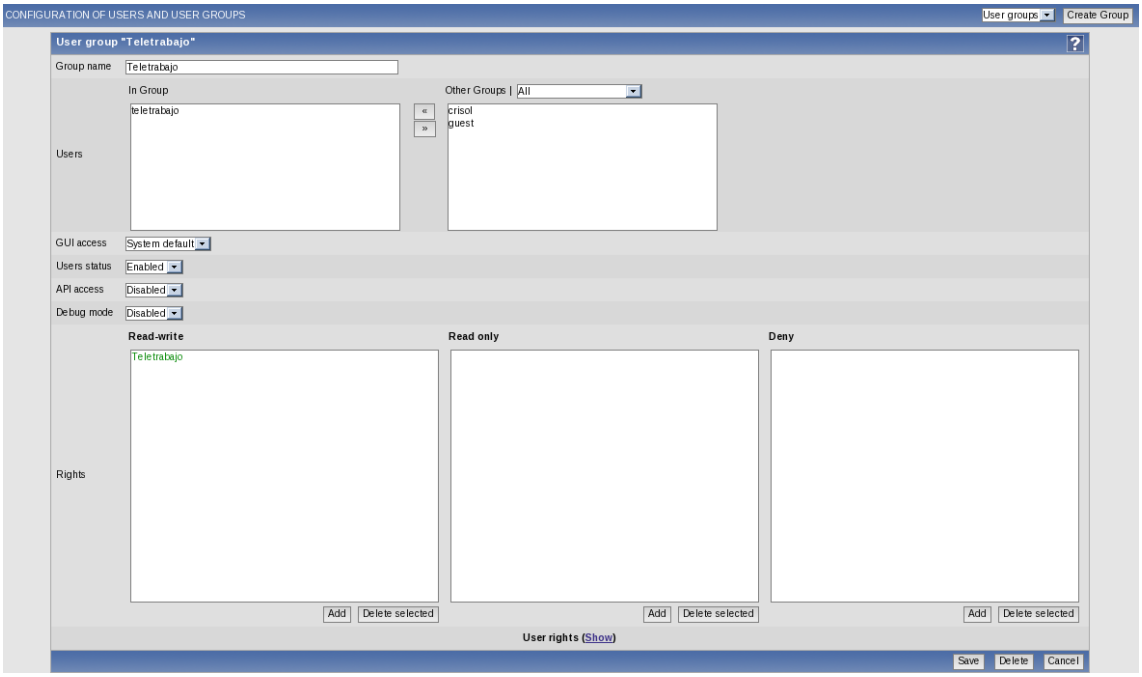

Ilustración 86 Pantalla de definición de rol en Zabbix

Una vez de vuelta a la pantalla de definición de rol se pulsa sobre “Save” para guardar los cambios y que quede definido el grupo. Aparecerá “Teletrabajo” en la pantalla de administración de usuarios, para agregar el usuario teletrabajo, se pincha sobre la palabra en verde “users” junto a Teletrabajo, y en la siguiente pantalla “Create user” para crear el usuario de la misma manera que se hizo anteriormente.

4.5.7.3. Definición de plantilla

En el punto *4.4.3 Diseño de la función de monitorización* se establecieron una serie de plantillas que tienen que ser definidas ahora, se explicará el proceso para definir una plantilla genérica, aunque el proceso es equivalente para el resto (excepto en el de aplicar herencia).

Para definir una plantilla, se accede a la pestaña “Configuration” y se selecciona el menú “Host groups”, a continuación, en la tabla mostrada se pincha sobre el “Templates” que está localizado en la fila OSL. Se nos presentará una pantalla con las plantillas de la OSL (vacía en este momento) y se pincha sobre “Create template”.

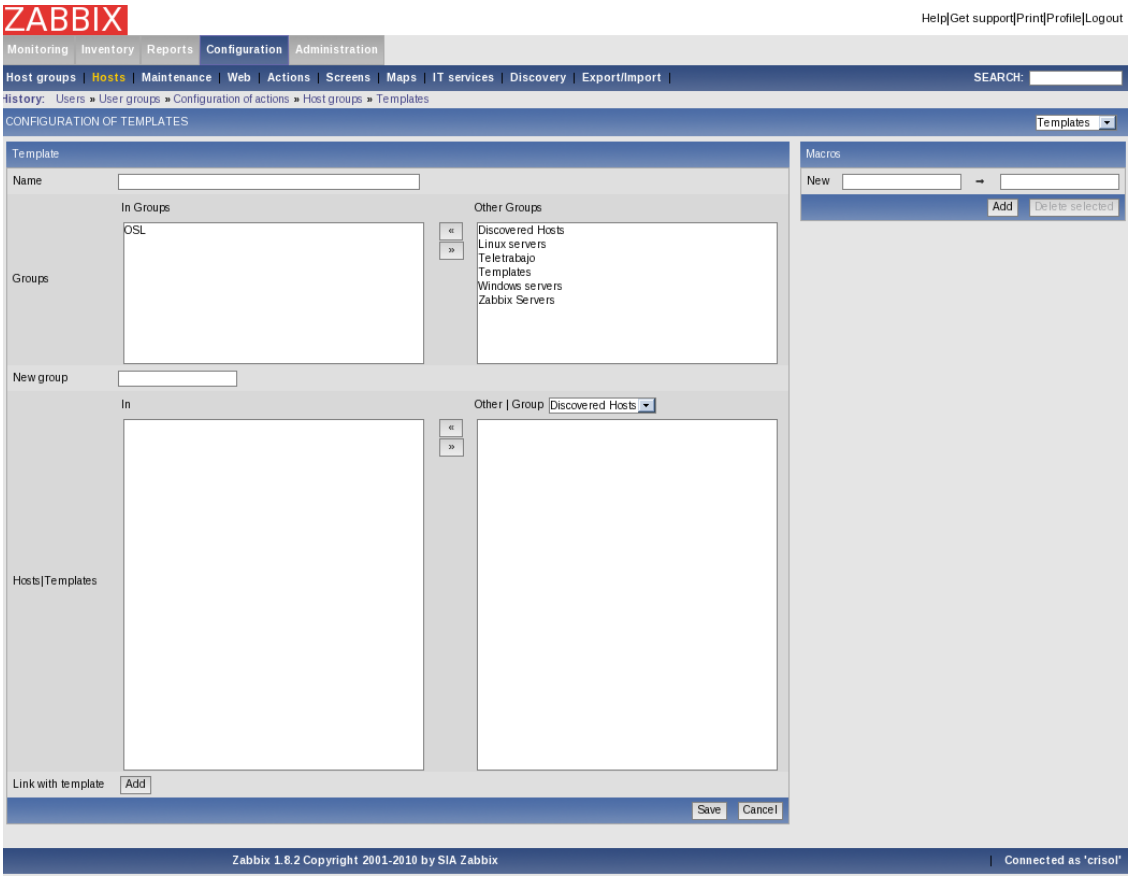

Ilustración 87 Pantalla de creación de plantilla en Zabbix

Para definir la plantilla, únicamente es necesario darle un nombre en el cuadro de texto con la etiqueta “Name” y pinchar sobre el botón “Save”. De esta manera se definen las plantillas “OSL Windows”, “OSL Windows Teletrabajo”, “OSL Linux”, “OSL Linux Server”, “OSL Linux Monitorización” y “OSL Linux Teletrabajo”. Para el caso de las plantillas herederas de otras, es necesario que antes de guardar la plantilla, se pinche sobre el botón “Add” junto a la etiqueta “Link with template”, lo que desplegará una lista de las plantillas del sistema. Se selecciona únicamente la plantilla padre directa y automáticamente la plantilla heredará todos los elementos definidos en el padre.

4.5.7.4. Definición de *items*

El procedimiento para definir un *item* en un host y en una plantilla es idéntico, ya que Zabbix trata de igual manera a las plantillas que los hosts. A continuación se explica cómo se define el *item* que comprueba el estado del servicio RDP en las máquinas virtuales de Teletrabajo, el procedimiento para crear todos los *item*s especificados durante el diseño es igual, pero dependiendo del elemento a medir se usan unas funciones y parámetros distintos, que pueden ser consultados en el manual de Zabbix (Zabbix SIA, 2011).

Para definir un *item* se accede a la interfaz de Zabbix y se selecciona la pestaña "Configuration" y el menú "Host groups". Aparecerá una lista con los grupos definidos, y las plantillas y host que tiene cada uno. Se pincha sobre "Templates" que está en la fila del grupo OSL y aparecerá la pantalla de plantillas del grupo OSL que se puede ver en *Ilustración 88 Pantalla de plantillas del grupo OSL*.

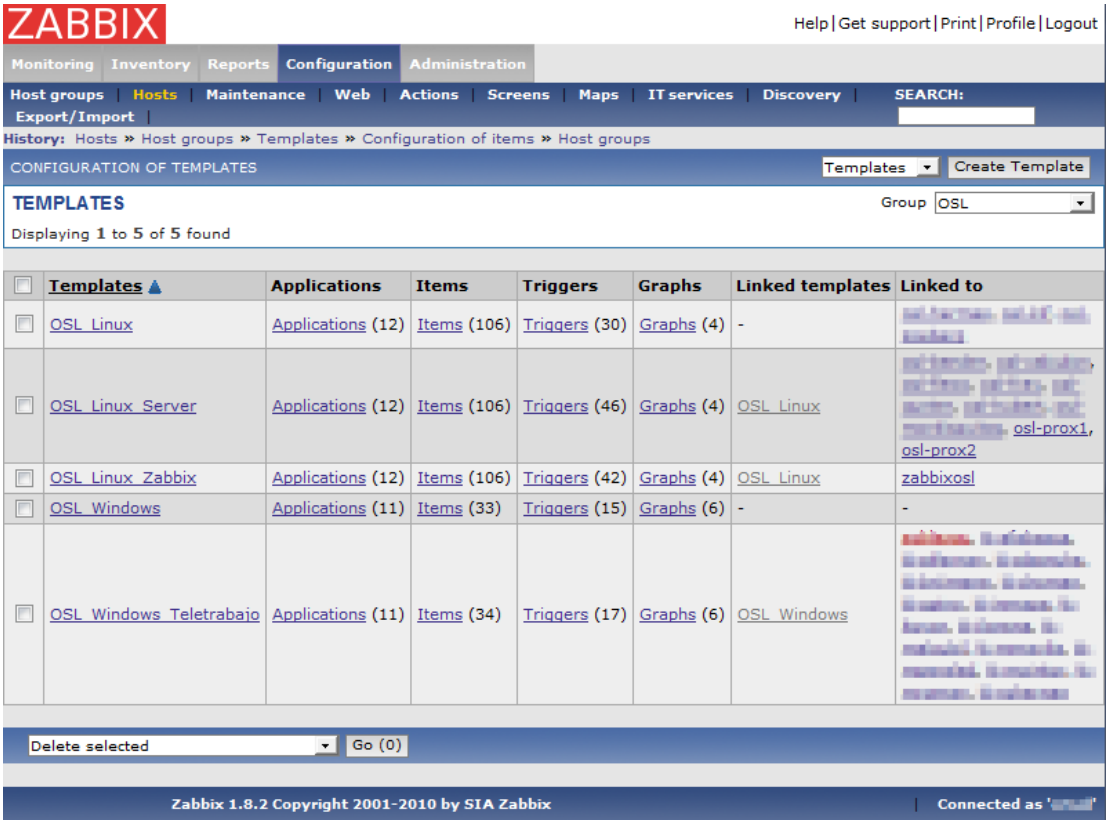

Ilustración 88 Pantalla de plantillas del grupo OSL

En esta pantalla pinchamos sobre la palabra "Items" que aparece en la fila de la plantilla OSL_Windows_Teletrabajo, haciendo que aparezca la pantalla de *item*s de un host (o plantilla).

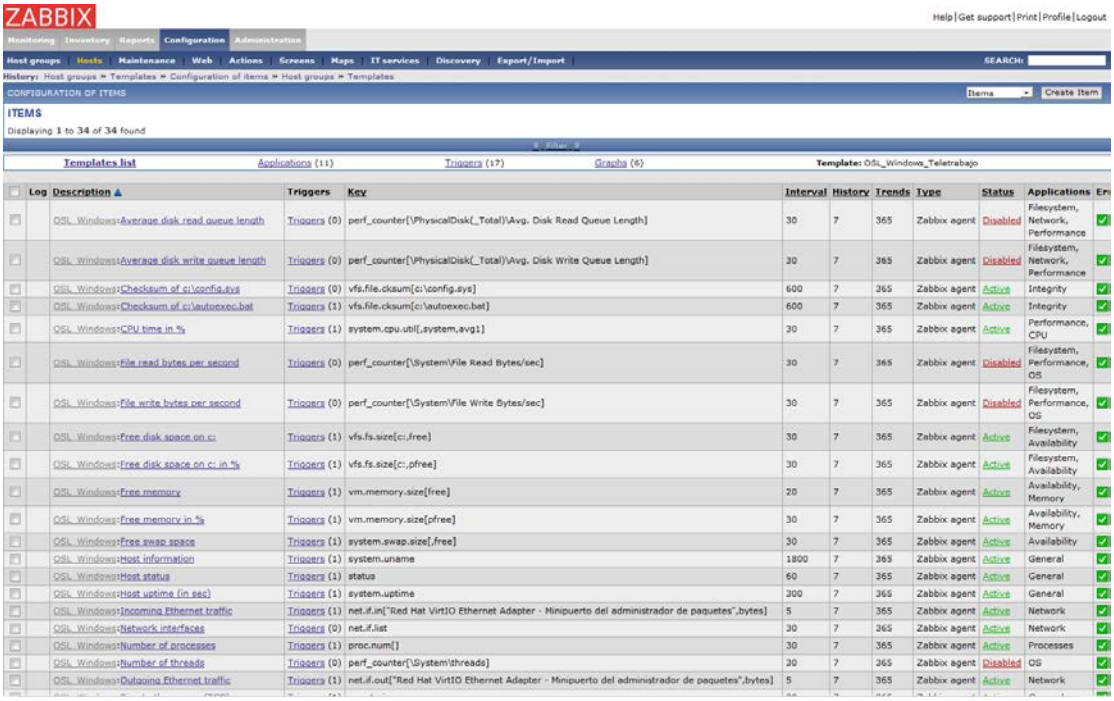

Ilustración 89 Pantalla de *items* de un host

En la pantalla de *items* de la plantilla OSL_Windows_Teletrabajo pinchamos en la esquina superior derecha en el botón “Create Item”, haciendo aparecer la pantalla de definición del *item*.

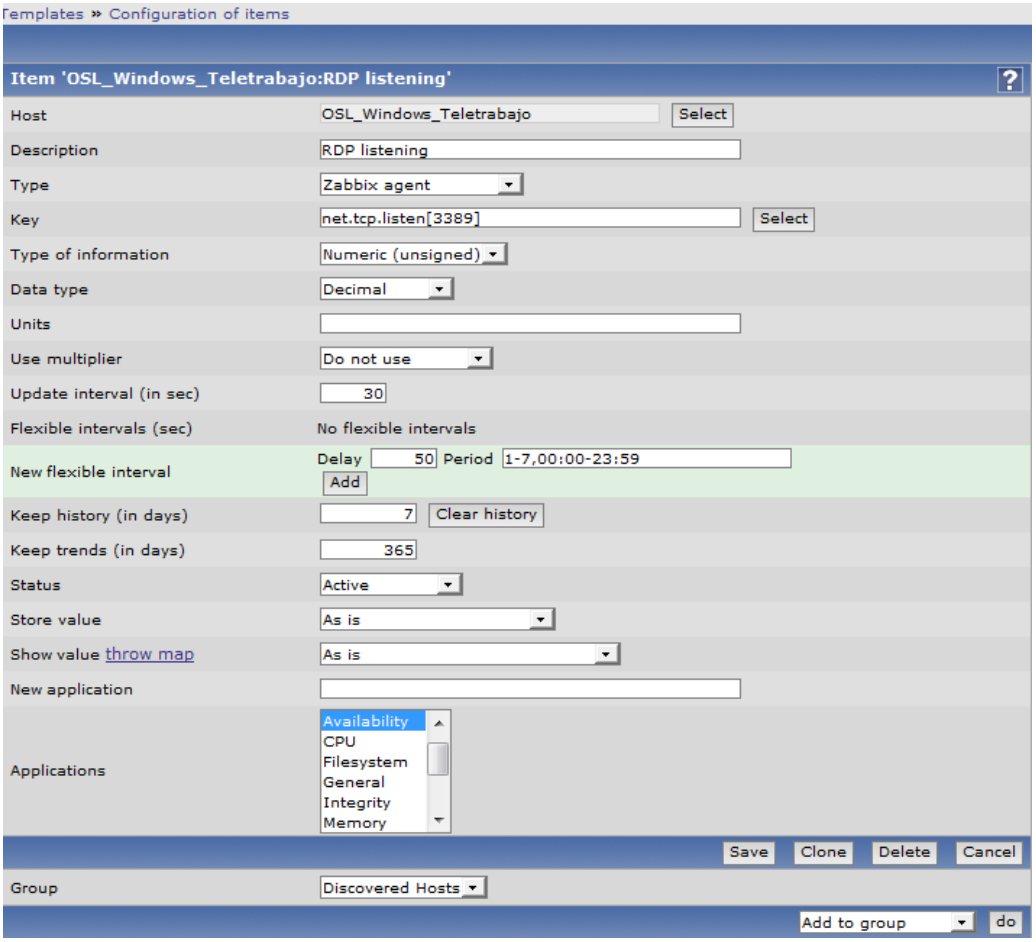

Ilustración 90 Pantalla de definición del *item*

Para crear el *item*, se le da un título que describa lo que hace en el campo "Description", es necesario poner un título descriptivo para saber qué hace un *item* en concreto, ya que el sistema los identifica mediante el valor del campo "Key" que no es muy descriptivo para seres humanos. En este caso como queremos medir que el host está escuchando conexiones RDP, llamamos al *item* "**RDP Listening**".

"Type" permite definir cómo funcionará este *item*, entre las opciones posibles esta que el agente envíe activamente el valor de este *item* al servidor, o que el servidor pregunte al agente por el valor del *item*. También se puede definir como una comprobación SNMP o, incluso, permite definir *item*s de tipo calculado en el que se aplican operaciones aritméticas a uno o varios elementos medidos a la vez. En este caso, es una comprobación por red que se encarga de realizar el servidor sin necesidad de que intervenga el agente, por ello el tipo se establece a "**Simple check**".

En el campo "Key" se escribe la sentencia con la función y los parámetros que define la medición a realizar, en el manual de Zabbix se pueden encontrar tablas con todos los keys disponibles y compatibles con cada sistema operativo, así como

el significado y valor que pueden tomar los parámetros. Además, pinchando sobre el botón "Select", se puede seleccionar el key deseado directamente de una lista, el problema es que no ofrece ninguna descripción, siendo recomendable consultar el manual para cada *item* para asegurarse de definirlo bien. Para averiguar si un servicio de red está activo, se puede comprobar si responde ese servicio en su puerto correspondiente, el key utilizado es el que hace conexiones *tcp* al puerto indicado, que en este caso es el de RDP que es el 3389 quedando la definición del key "**net.tcp.listen[3389]**".

Los datos que devuelve este key son booleanos: o está a la escucha el servicio o no lo está. Al no haber un tipo de datos booleano en Zabbix, indicamos en "Type of information" que es un número natural (0 falso y 1 verdadero) seleccionando del desplegable "**Numeric (unsigned)**".

En el campo "Data type" se establece la base numérica en la que se representa el dato, dándonos a elegir entre números octales, decimales o hexadecimales. Puesto que los valores que obtendremos del key son binarios, es indiferente que opción se escoja, por lo que se deja en "**Decimal**" que es valor por defecto.

La medición que afecta a este *item* es un valor booleano que no tiene unidades, tanto se deja vacío el campo "Units". En caso de que la medición se haga en bytes, porcentajes o cualquier otra unidad se ha de indicar en este campo.

Si deseásemos multiplicar el valor del *item* y almacenar este valor modificado, se selecciona la opción "Custom multiplier" en el desplegable "Use multiplier", pero en este caso no es necesario al tratar con valores booleanos, de modo que se establece en la opción "**Do not use**".

En "Update interval" se especifica cada cuanto tiempo deseamos que se mida el *item* en el host, cada medición se almacenará en la base de datos, por lo que no es conveniente usar intervalos cortos para todo tipo de *item*s En caso de un *item* crítico, como es este, hacemos que se mida cada poco tiempo, por ejemplo "**30**" segundos. En cambio, para *item*s que midan la versión del núcleo de Linux o la capacidad del disco duro se pueden usar intervalos mucho más grandes de media hora y horas, ya que son elementos muy estáticos y por lo general poco críticos.

Se puede indicar la vida de cada valor del *item* en la base de datos mediante "Keep history" y "Keep trends". "Keep history" sirve para indicar cuantos días se almacenará un valor determinado en la base de datos. Cada cierto tiempo, Zabbix realiza operaciones de limpieza en la base de datos, eliminando todos los valores que hayan excedido este valor. Por homogeneidad, en todos los *item*s hemos definido que se almacenen sus valores durante **7** días, por lo que siempre podemos

consultar todo el historial de valores de un *item* en una semana. Respecto a "Keep trends", no almacena los valores en sí, sino que almacena estadísticas de los *item*s, como puede ser el valor mínimo que ha tenido, su valor máximo o su valor medio en un rango de tiempo. De esta manera se requiere mucho menos espacio en la base de datos para almacenarlos y permite realizar gráficos con intervalos de tiempo mucho mayores, por esto establecemos un valor grande para todos los *item*s de **365** días.

"Status" sirve para activar o desactivar el *item*, de forma que podamos indicar manualmente si queremos que sea medido el *item* en cuestión en un host. En general todos los *item*s en las plantillas los establecemos con el valor "**Active**", pero al ser aplicada la plantilla en un host concreto, es posible que sea incapaz de medir ese *item* en el host, cambiándose este estado automáticamente al estado "Not suported" y generando en consecuencia mensajes de error en el log. Es por tanto aconsejable, después de aplicar una plantilla, revisar los logs y comprobar que todos los *item*s están funcionando, desactivando aquellos que no funcionen.

Por último se pueden definir aplicaciones, que agrupan un conjunto de *item*s con un objetivo de monitorización común, como pueden ser rendimiento del equipo, servicios ofrecidos, capacidades etc. En una lista se muestran las aplicaciones que ya están definidas en el sistema. En este caso se está midiendo la disponibilidad de un servicio, por lo que se escoge la aplicación "**Availability**" en "Applications". Se pueden escoger varias aplicaciones a las que pertenecerá un único *item* y, mediante "New application", se puede crear una nueva aplicación a la que pertenecerá el *item*.

Cuando se hayan definido todas las características del *item*, se pulsa sobre "Save" y el *item* quedará registrado en el sistema.

4.5.7.5. Definición de *triggers*

La creación de *trigger*s se realiza en la pantalla de *trigger*s de un host, a la cual podemos acceder desde la pantalla de plantillas del grupo OSL (*Ilustración 88 Pantalla de plantillas del grupo OSL*). Continuando con el *item* definido en el punto anterior, a continuación se explicará cómo se define un *trigger* que informe cuando deja de estar disponible el servicio RDP. Aunque en este caso un *trigger* parametriza y vigila un *item*, no tiene por qué ser así, ya que un *trigger* puede consultar varios *item*s o incluso, en casos excepcionales, ninguno.

En la pantalla de plantillas se pincha en “Triggers”, en la fila de la plantilla “OSL_Windows_Teletrabajo”, dirigiéndonos a la pantalla de *trigger*s.

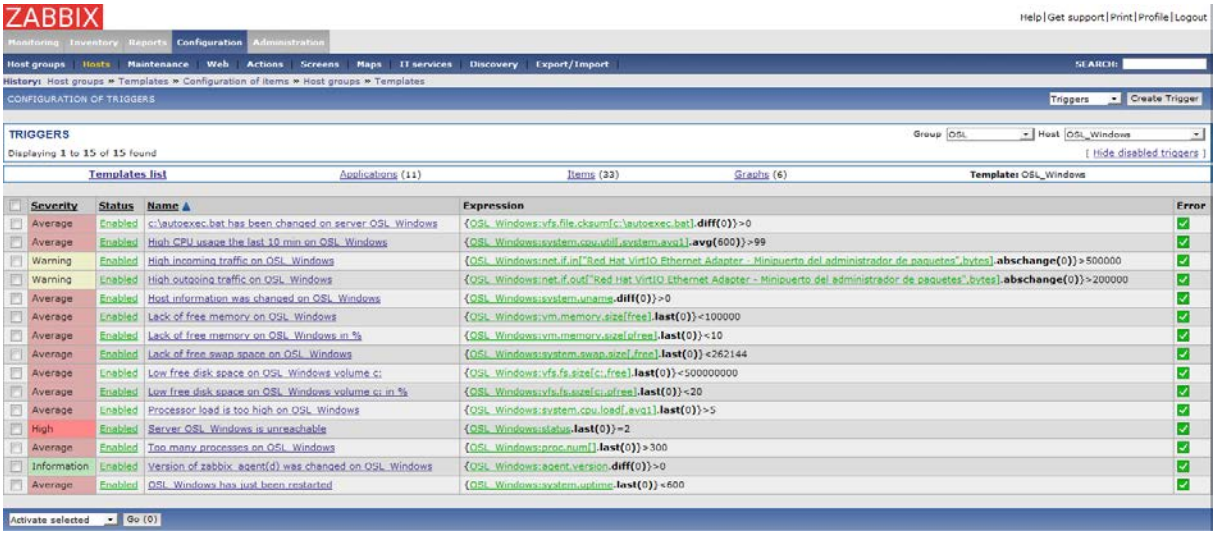

Ilustración 91 Pantalla de *trigger*s de un host

La pantalla de *trigger*s (*Ilustración 91 Pantalla de triggers de un host*) dispone en una tabla todos los *trigger*s de un host, indicando si están activos o no, la gravedad, elnombre que tienen y la condición mediante la cual se activan. Para añadir uno nuevo, se pincha sobre el botón “Create Trigger” que nos lleva a la pantalla de definición de *trigger*.

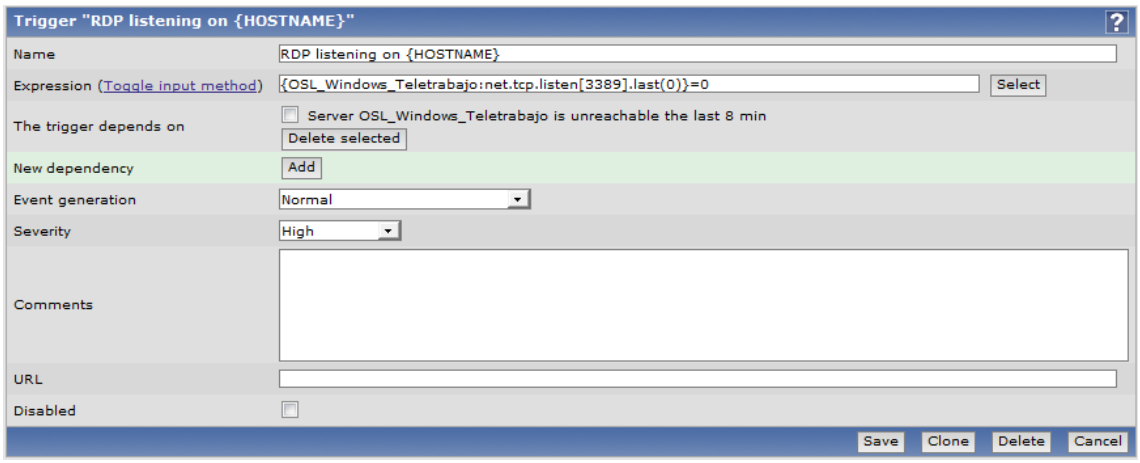

Ilustración 92 Pantalla de definición de *trigger*

En primer lugar se pone el nombre al *trigger* en el campo “Name”, este nombre es el que se le mostrará al administrador, tanto en las pantallas de resumen de estado de los equipos como en las notificaciones. Se establece el nombre “**RDP listening on {HOSTNAME}**” para indicar que mientras el

trigger este en estado "OK" significa que el servicio RDP está a la escucha, en el host {HOSTNAME}. {HOSTNAME} es una macro que se remplaza automáticamente por el nombre del host en el que se está monitorizando la alerta, por lo que resulta muy interesante para definir el nombre en las plantillas con macros. De esta forma, cuando sean aplicadas a los host, cada uno tendrá los *trigger*s personalizados con el nombre del host, siendo así fácil conocer dónde ha surgido un problema, en caso de que se envíe una notificación.

La condición lógica que se debe cumplir para que cambie el estado del Trigger de "OK" a "Problem" se define en el campo "Expression". Si pinchamos sobre el botón "Select", aparece un diálogo para definir la expresión lógica en el que se deben definir tres partes. La primera de las partes "Item" es el *item* que se comparará para determinar el estado del *trigger*. Para seleccionar cuál se utilizará, aparece otro botón "select" que abre un nuevo diálogo en el que, tras fijar un grupo y un host (o plantilla) de ese grupo, se presenta una lista con todos los *item*s definidos en ese host (o plantilla) para poder seleccionar uno. Se selecciona "**OSL_Windows_Teletrabajo:RDP listening**". La segunda parte "Function", es un desplegable en el que se elige el operador lógico que se aplicará en la condición del *trigger*. Permite seleccionar operadores del tipo: ultimo valor menor que, últimos N valores iguales a, o suma de los últimos N valores mayor que. Puesto que el *item* seleccionado es booleano, queremos un operador que permita detectar un cambio de valor, seleccionando para ello "**Last value = N**". Finalmente, el tercer elemento "N" permite establecer una constante con la que se comparará la expresión. Puesto que, como se ha dicho anteriormente, el valor "**0**" es equivalente a falso, se establece ese valor. Para introducir la expresión terminada se pulsa el botón "Insert" que cerrará el diálogo.

Sólo tiene sentido emitir alertas por este *trigger* si el host al que pertenece está funcionando correctamente, de tal manera que si el host no está disponible todos los servicios que ofreciera tampoco estarán disponibles. Definiendo una dependencia se consigue que el *trigger* sólo este activo mientras otro *trigger* este en estado "OK". En este caso, depende de que el host esté disponible. Para añadir la dependencia se pincha sobre el botón "Add" junto a la etiqueta "New dependency". Se mostrará un diálogo con todos los *item*s definidos en un host (o plantilla) de un grupo. Se fija el grupo a OSL en el desplegable y, en el siguiente desplegable, se selecciona la plantilla OSL_Windows_Teletrabajo, haciendo aparece los *trigger*s de esa plantilla. Se selecciona el *trigger* "**Server OSL_Windows_Teletrabajo is unreachable**" y se confirma con el botón "Select".

Por último, se define la gravedad del *trigger* seleccionándola en el desplegable "Severity" que ofrece cinco niveles de alerta: desde la menos preocupante "Information", hasta la más grave "Disaster". En caso de que se active el *trigger*, se mostrará un color en función de la gravedad que se haya definido. En este caso, que no funcione el servicio RDP es un problema importante, pues no permite a los teletrabajadores usar el servicio, pero no es un problema que implique una pérdida de información o algo irreparable, por lo que lo establecemos a "**High**".

Si deseásemos que no se verificase este *trigger* marcaríamos la casilla "Disable" en el host en el que queramos desactivarlo. Tendría gran utilidad en el caso de aplicar la plantilla OSL_Linux_Server, en la que se definen *trigger*s para controlar servicios como NTP o SNMP que no tienen por qué estar presentes en todos los servidores, de forma que activando esta casilla, se dejan de emitir alertas por estos servicios. En este caso se deja desmarcada.

4.5.7.6. Definición de notificaciones

Las notificaciones en Zabbix se definen como acciones. Las acciones no se limitan únicamente a enviar mensajes cuando se activa un *trigger*, sino que permite programar pasos a ejecutar. Por ejemplo se han definido a acciones que lancen el demonio sshd en caso de que deje de estar en ejecución.

Para definir la notificación, dentro de la interfaz de Zabbix, se selecciona la pestaña "Configuration" y el sub-menú "Actions".

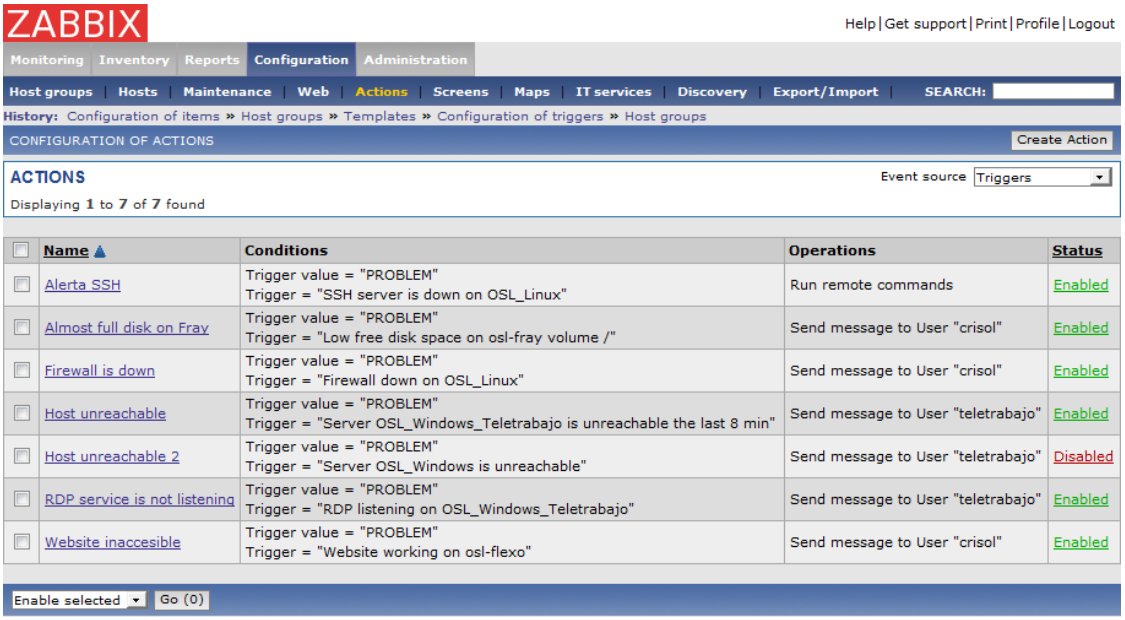

Ilustración 93 Pantalla de acciones

En la pantalla de acciones, se muestran todas las acciones definidas en el sistema, independiente de en qué hosts, plantillas o grupos sean usados, es decir, las acciones no se clasifican: todas pertenecen al sistema. Siguiendo la temática del *item* y del *trigger*, ahora se explicará el procedimiento para definir una notificación que envíe un mensaje al administrador cuando deje de estar activo el servicio RDP en las máquinas virtuales de Teletrabajo.

Para definir la notificación, se pincha con el ratón sobre el botón "Create action", abriendo así la pantalla de definición de la acción.

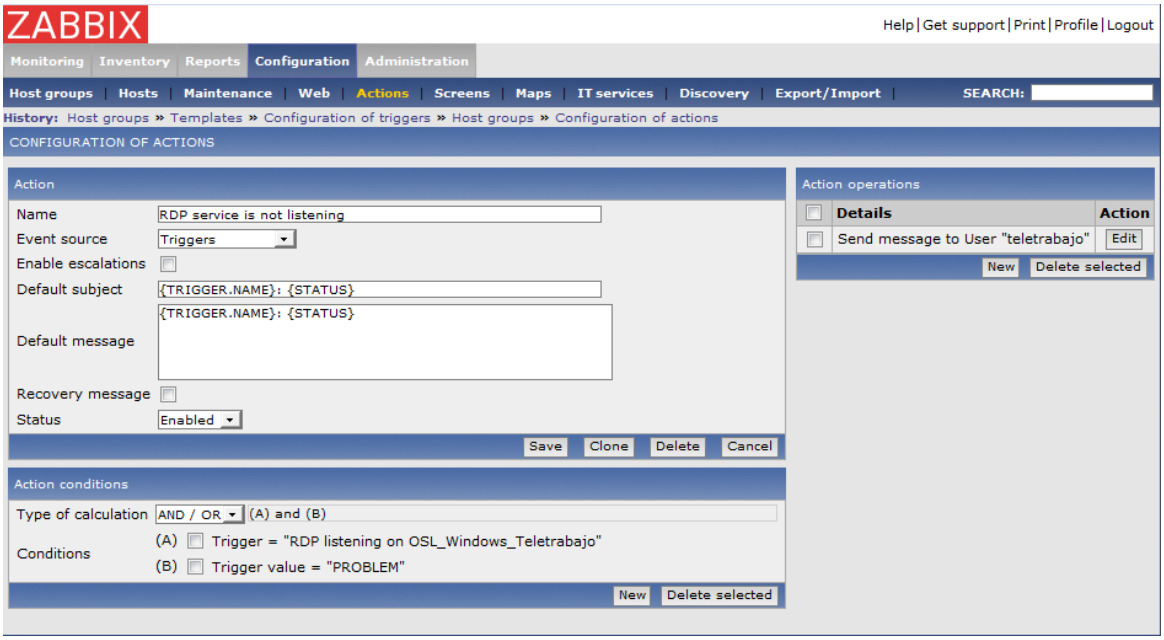

Ilustración 94 Pantalla de definición de acción

En "Name" se pone el nombre que se le va a dar a la acción, en este caso "**RDP service is not listening**", al ser la acción que se ejecutará cuando deje de escuchar el servicio RDP.

En "Event source" se deja el valor "Triggers" para indicar que esta acción será activada por *trigger*s.

El asunto del mensaje que se enviará se define "Default subject" y el cuerpo del mensaje en "Default message". Ambos campos tienen como valor por defecto un mensaje compuesto por dos macros que dejamos así: "**{TRIGGER:NAME}: {STATUS}**". La primera macro toma el valor del *trigger* implicado en la notificación, para que el administrador sepa cuál es el problema y la segunda macro, toma el valor del estado del *trigger*, es decir "OK" o "PROBLEM".

"Status" permite indicar si queremos que se realice o no la acción definida, permitiendo desactivarla cuando por algún motivo no es necesario que se envíen

notificaciones o no deseamos que levante servicios automáticamente, como puede ser el caso de paradas del servicio por mantenimiento. Para que se envíen las notificaciones dejamos el valor "**Enabled**" que pone por defecto.

En la parte inferior hay un cuadro llamado "Action conditions", en el que especificamos la condición lógica que se debe evaluar para determinar si se activa o no la acción. En primer lugar se activa el botón "New" de este cuadro, haciendo aparecer un cuadro nuevo denominado "New condition". La condición que queremos definir es referente a un *trigger* determinado, por lo que se selecciona en el primer desplegable "Trigger". Automáticamente, el segundo desplegable cambiará al valor "=" que dejamos así. La tercera parte de la condición no puede ser escrita a mano, ya que espera un *trigger*. Para rellenarlo se pincha con el ratón sobre el botón "Select" que hará aparecer una pantalla en la que, tras seleccionar un grupo (OSL) y un host o plantilla (OSL_Windows_Teletrabajo), lista todos los *trigger*s definidos en ese host. Se busca el *trigger* creado en el punto anterior "**RDP listening on OSL_Windows_Teletrabajo**" y se selecciona. Finalizamos la edición de la condición pulsando sobre "Add", haciendo que en el cuadro "Action conditions" aparezca una condición etiquetada como "(A)" con la siguiente condición "**Trigger = "RDP listening on OSL_Windows_Teletrabajo"**". Ahora hay que definir, cuál de los dos valores del *trigger* seleccionado es el que queremos que sea notificado, siendo necesario para ello crear una segunda condición (B), de la misma manera que la anterior, sólo que en el primer desplegable se seleccionara "Trigger value", en el segundo "=" y el tercero se convertirá en un desplegable en el que se selecciona "PROBLEM", quedando definida la condición "(B)" "**Trigger value = "PROBLEM"**". En el desplegable "Type of calculation" se escoge "and", de manera que para que se active la acción se deben de cumplir las dos condiciones. Así que para que se active la condición debe haber en el host un *trigger* denominado "RDP listening on {HOSTNAME}" y el valor de ese *trigger* debe ser "PROBLEM".

Finalmente, se ha de indicar que se envíe una notificación cuando se cumpla la condición y cómo se enviará la notificación al administrador. Para especificarlo hay un cuadro a la derecha llamado "Action operations". Para definir la nueva operación se pulsa sobre "New", que hará aparecer un nuevo cuadro "Edit operation". El primer desplegable "Operation type" permite indicar cuál de las dos acciones es la que se desea realiza: enviar una notificación o ejecutar un comando remoto. Seleccionamos "Send message" para enviar mensajes. El siguiente paso es introducir a quién queremos que se envíe el mensaje, pudiendo hacerlo a un usuario en concreto o a un rol. En este caso queremos que la notificación la reciba

el administrador de Teletrabajo, así que se selecciona la opción "Single user" y, tras pulsar sobre "Select", aparecerá una lista con los usuarios definidos en el sistema de monitorización. Se selecciona el usuario "teletrabajo" y aparecerán en la parte inferior todos los medios de comunicación que hay definidos para ese usuario (correo electrónico, Jabber o SMS). Podemos especificar a cuál de los medios se enviará la notificación en el desplegable "Send only to" pero, puesto que el usuario teletrabajo sólo tiene definido una dirección de correo electrónico, dejamos seleccionado "-All-" para que se envíe la notificación al correo electrónico por ahora. Si en el futuro se introduce otro medio, también sea enviada a través de él.

Para finalizar la edición de la acción, se selecciona "Add" para que se quede almacenada la operación y "Save" para guardar los cambios de la acción.

4.5.7.7. Definición de gráfica

Para la creación de gráficas hay que acceder a la pantalla de plantillas de la interfaz de Zabbix y seleccionar la palabra "Graphs" que hay en la fila de la plantilla en la que se quiere definir la gráfica. En este caso queremos definir una gráfica que muestre el porcentaje de espacio libre en el disco duro que tienen los equipos GNU/Linux, por ello se selecciona el "Graphs" que hay en la fila de OSL_Linux, lo que hará aparecer la pantalla de gráficas de la plantilla.

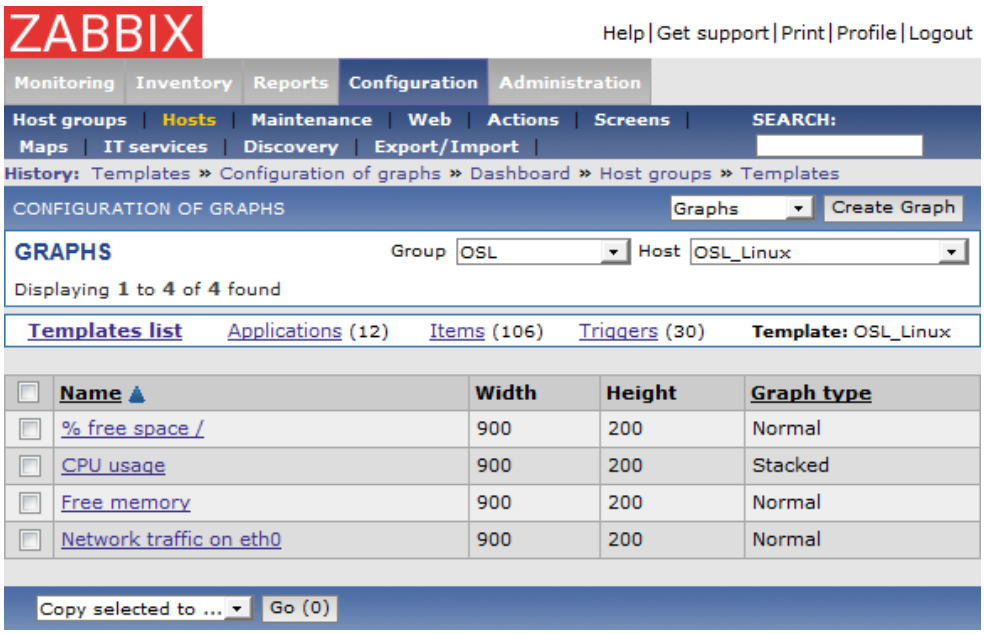

Ilustración 95 Pantalla de gráficas de un host

Para crear una nueva gráfica en esta plantilla hay que pulsar el botón "Create Graph", para que aparezca la pantalla de edición de gráfica.

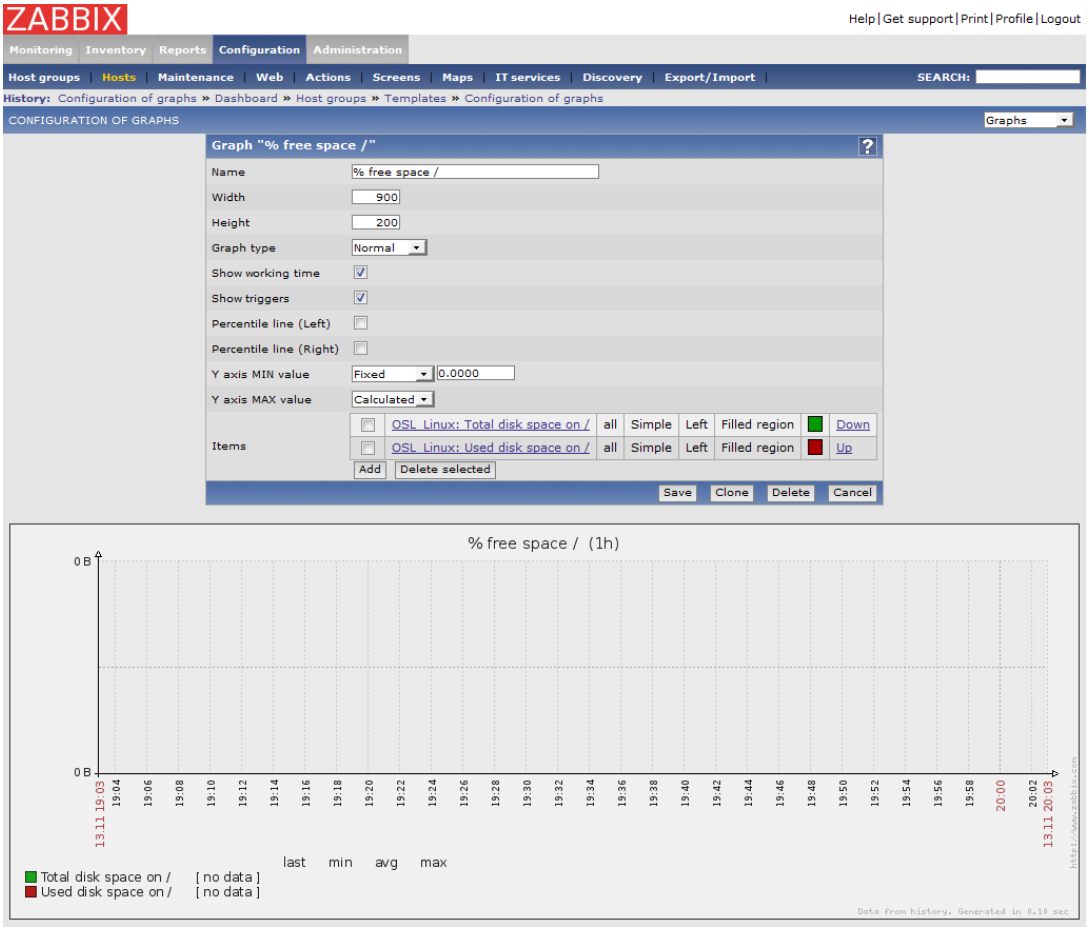

Ilustración 96 Pantalla de edición de gráfica

Primero se establece un nombre en el campo "Name" que utilizará el sistema como identificador de la gráfica. Puesto que la gráfica mostrará el porcentaje de espacio libre en la partición "/" del disco duro, se asignará el nombre "**% free space /**".

Para que todos las gráficas presenten el mismo aspecto, se dejan todos los parámetros por defecto, excepto "Graph type", que es donde se indica que tipo de gráfica se puede crear. En este caso "Normal", para disponer en una línea temporal todos los valores que han tomado los *items*.

Para agregar un *item* en la gráfica, se pincha con el ratón sobre el botón "Add" para desplegar el diálogo "Graph item". Para seleccionar un *item* se pulsa sobre el botón "Select" del cuadro "Parameter", que hará aparecer una lista con todos los *items* definidos en el host. Seleccionamos el *item* "**Used disk space on /**". En el desplegable "Draw stile" se puede decir cómo deseamos que se dibuje el

item en la gráfica, pudiendo escoger que se dibuje como puntos, líneas, gradientes o un área. Esta última es la opción deseada, por ello se establece su valor a "Filled región". Por último se establece un color rojo para el área que define el *item* en el campo "Colour", pinchando con el ratón sobre su cuadro de texto y seleccionando el color rojo de la paleta de colores que se despliega. Terminado de definir el *item* en la gráfica, se pulsa el botón "Add" para agregarlo a la gráfica.

Se añade un segundo *item* en la gráfica, igual que se ha hecho con el anterior, pero esta vez seleccionando el *item* "**Total disk space on /**" y poniéndolo de color verde. Para que el área que define este *item* quede al fondo de la gráfica, parcialmente cubierto por el área del *item* de espacio adecuado, y hacer que el área mostrada sea equivalente al espacio libre, se selecciona "Up" en el *item* para alterar el orden en el que son dibujados, hasta que quede el primero de la lista.

Por último para que quede registrada la gráfica definida se pulsa sobre el botón "Save".

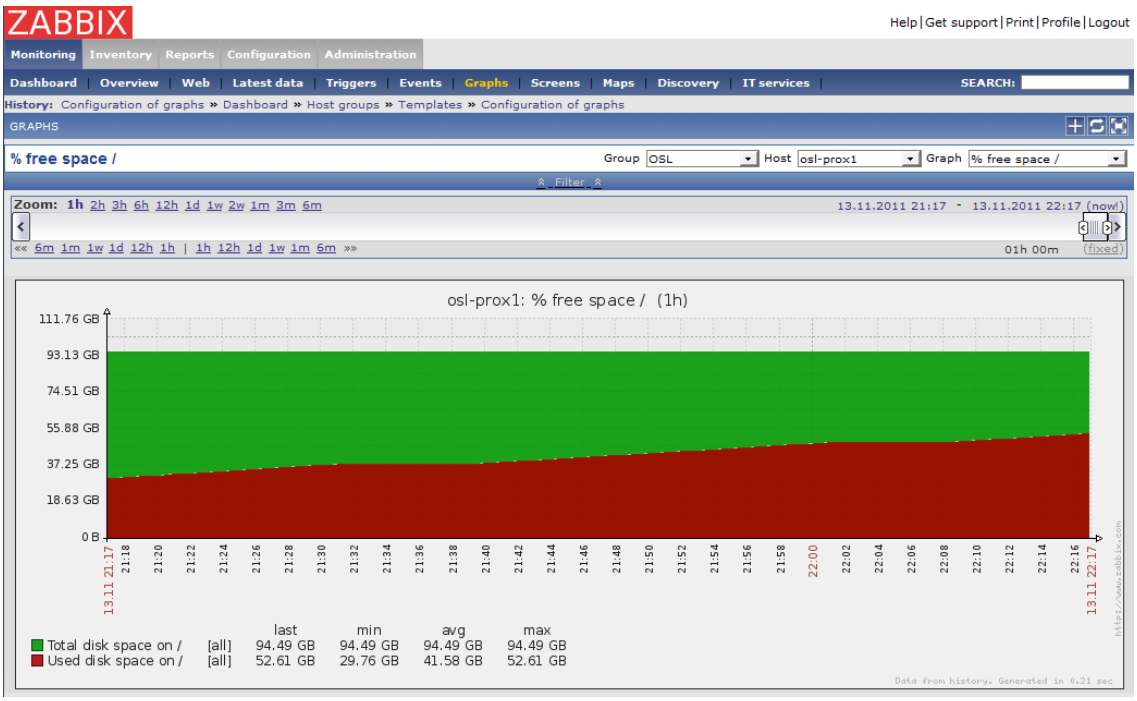

Ilustración 97 Gráfica "% free space" en "OSL_Linux"

4.5.7.8. Agregar hosts y aplicar plantillas

Antes de agregar un nuevo host que monitorizar en el servidor de Zabbix, hay que haber instalado el agente de Zabbix en el equipo, haberlo configurado y

ejecutado según las instrucciones del punto *4.5.6 Configuración de agentes de monitorización.*

Para agregar el nuevo host, accedemos a la interfaz de Zabbix y se selecciona la pestaña "Configuration", de entre los menús que aparecen se selecciona "Hosts" y se nos presentará la pantalla de hosts.

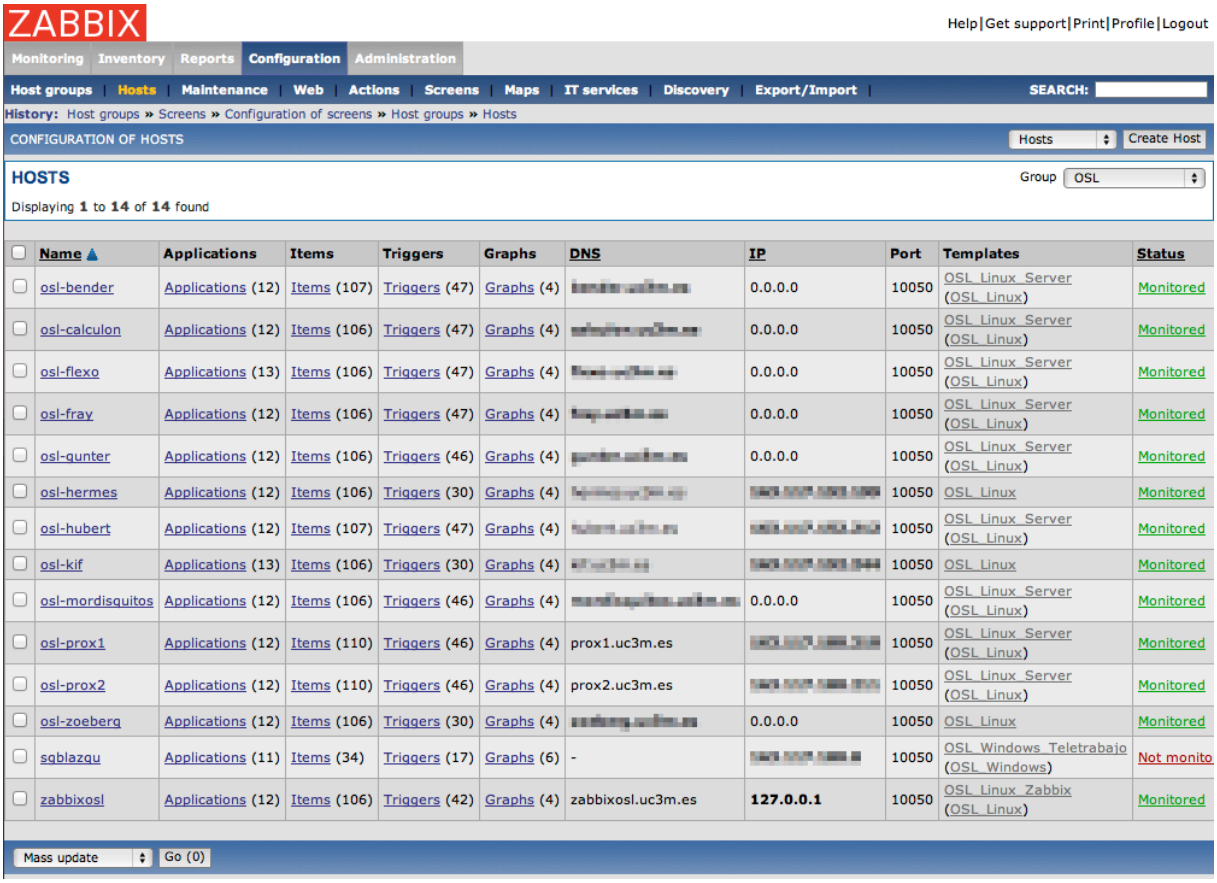

Ilustración 98 Pantalla de hosts

En la esquina superior derecha de la pantalla de hosts hay un botón llamado "Create hosts", al pulsarlo hará que aparezca la pantalla de configuración del host.

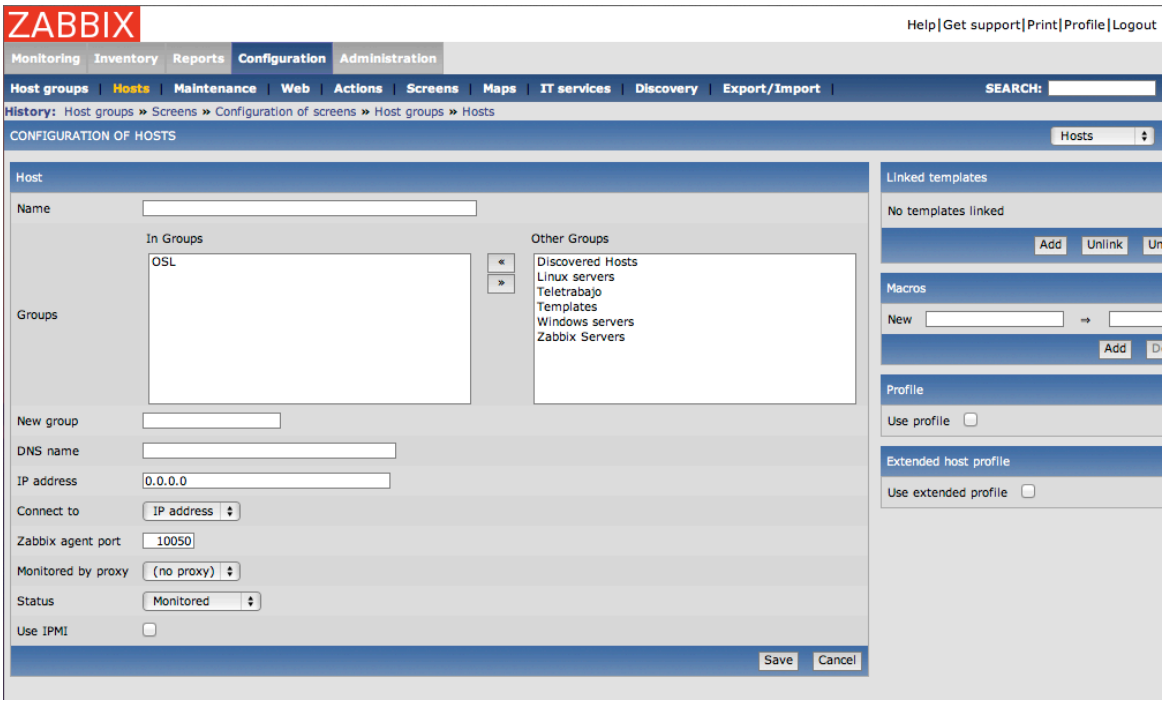

Ilustración 99 Pantalla de configuración del host

En primer lugar le asignamos el nombre que identificará al host dentro del servicio de monitorización, en el cuadro de texto etiquetado como "Name". En este caso la máquina que se va a agregar es "**osl-prox1**". Es importante que el nombre dado al host sea el mismo que se definió en el fichero de configuración del agente zabbix_agend.conf, en el parámetro "HOSTNAME".

Debajo del nombre aparecerá una etiqueta "Groups" con dos listas, en la lista de la derecha están todos los grupos de hosts definidos en el sistema, y a la izquierda, los grupos en los cuales se incluirá el nuevo host. El host que se va a agregar es el primer nodo del servidor de Teletrabajo, que es un equipo administrado por la Oficina de Software Libre, por tanto, buscamos el grupo "OSL" en la lista de la izquierda y pulsamos sobre el botón "<<", comprendido entre ambas listas, para que "OSL" pase de la lista de la derecha a la de la izquierda.

Ahora hay que indicar la dirección del host para que el servidor pueda ponerse en contacto con ella. Hay dos opciones para introducir la dirección de la máquina: usando su dirección IP o el nombre DNS que se le haya dado. De modo que en el campo "DNS name" se introduce el nombre "**prox1.uc3m.es**" y en el campo "IP address" la dirección IP de la máquina. A pesar de introducir la dirección de dos formas distintas, Zabbix sólo utilizará una de las dos (se introducen las dos a modo de anotación de correspondencias entre nombres e IPs). Para indicar qué dirección se quiere utilizar, se selecciona cualquiera de las dos

opciones en el menú desplegable "Connect to", en este caso seleccionamos "**IP address**" para evitar que tenga que resolver nombres.

En el desplegable "Status" podemos monitorizar o dejar de monitorizar un host en cualquier momento, por ejemplo, en caso de que el host vaya a dejar de estar en servicio un periodo de tiempo y no queremos recibir notificaciones referentes a él. Para que monitorice el nuevo host, se establece su valor a "**Monitored**".

Por último, queda aplicar una plantilla al host para que reciba todos los *item*s, *trigger*s y gráficos definidos en la plantilla, y así tener elementos que monitorizar. Para ello, a la derecha hay un cuadro denominado "Linked templates" con un botón "Add", al pulsarlo aparecerá un diálogo con plantillas definidas. Seleccionamos el grupo donde están definidas "OSL" en el menú desplegable "Group" y la plantilla "OSL_Linux_Server" en la lista, para aplicarla pulsando en el botón "Select".

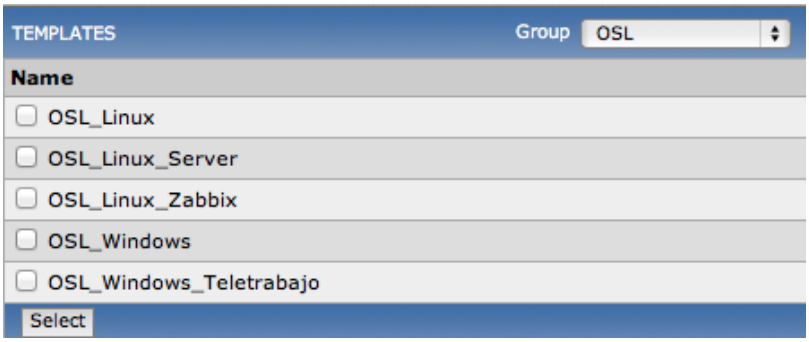

Ilustración 100 Pantalla de selección de plantilla

Tras volver a la pantalla de configuración del host, pulsamos sobre "Save" para comenzar a monitorizar el host.

4.5.7.9. Definición de pantallas

Las pantallas de Zabbix permiten visualizar en una única página varios elementos gráficos del servidor de monitorización, como pueden ser: gráficas, esquemas de red o resúmenes de *trigger*s. En el caso de este proyecto, se utilizarán para mostrar en una única página todas las gráficas definidas en cada máquina, y así poder ver rápidamente como se están comportando, en cuanto a rendimiento, los hosts (en especial las máquinas virtuales).

Para la creación de una pantalla, se ha de seleccionar la pestaña "Configuration" de la interfaz de Zabbix y, a continuación, el menú "Screens" haciendo aparecer la pantalla "screens" de Zabbix.

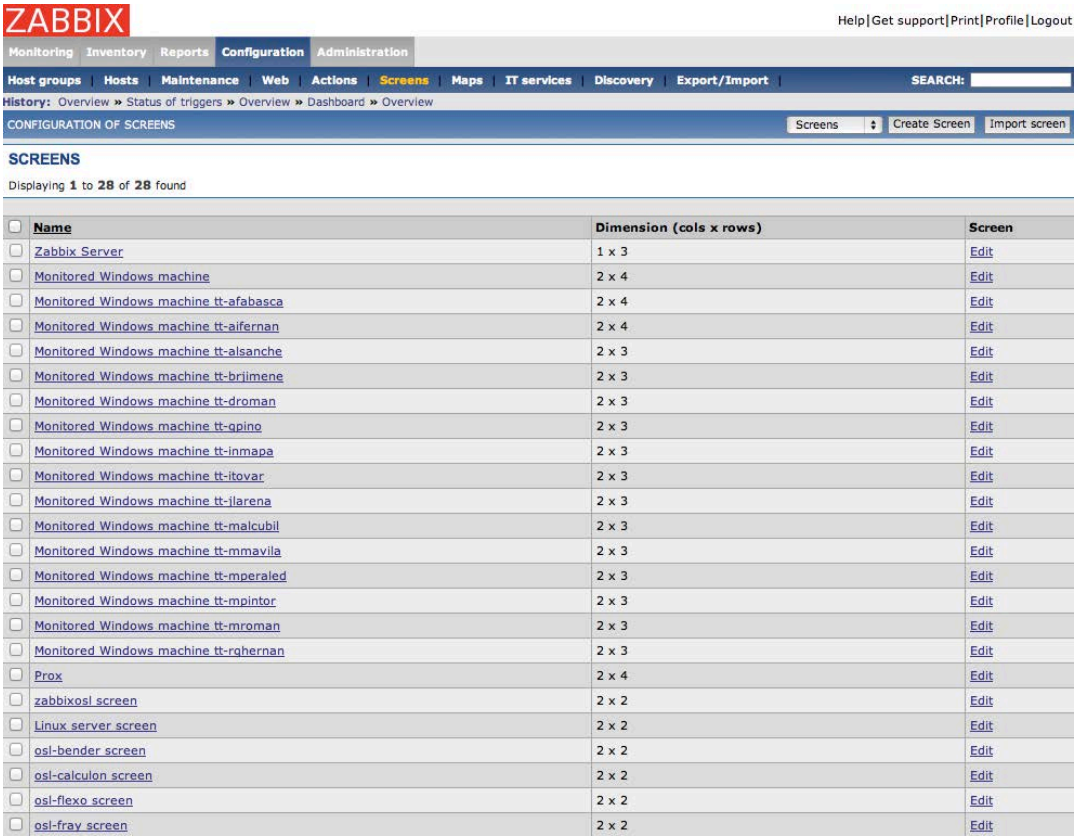

Ilustración 101 Pnatalla de screens

A continuación se pulsa sobre el botón "Create screen" para acceder a la pantalla de configuración de la pantalla.

Ilustración 102 Pantalla de definición de screen

En la pantalla de configuración se establece un nombre en el campo "Name" con el que será identificada. En este caso, puesto que se creará una pantalla que muestre todas las gráficas del clúster de Teletrabajo, se nombrará la pantalla "**Prox**". Los hosts de las plantillas para equipos GNU/Linux tienen cuatro

gráficas y, puesto que hay dos servidores, dispondremos las cuatro gráficas de cada servidor en una columna distinta. Por tanto, se pone un valor "**2**" en el campo "Columns" y "**4**" en el campo "Rows".

Una vez se pulse sobre el botón "Save", se volverá a la pantalla "screens" donde aparecerá la nueva pantalla "Prox", creada pero sin ningún elemento definido. Tras pinchar en ella, se accede a la pantalla de edición donde se podrán añadir los elementos a la pantalla.

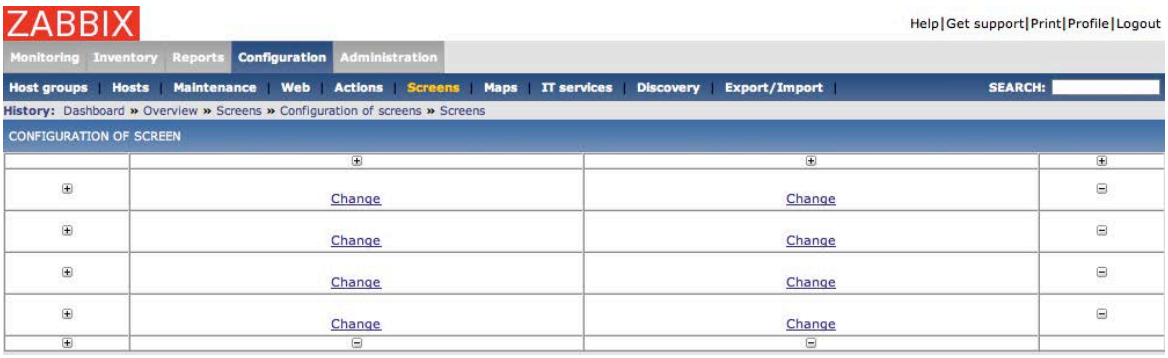

Ilustración 103 Pantalla de edición de screen

Para añadir un elemento a la pantalla, hay que seleccionar la palabra "Change" del cuadrante en el que se va a añadir la gráfica. En este caso, el de la esquina superior izquierda. En el recuadro seleccionado aparecerá el cuadro de configuración de la celda, en este cuadro, en el desplegable "Resource", se selecciona "Graph" para indicar que se insertará una gráfica. Al seleccionarlo, aparecerá justo debajo el cuadro "Graph name", al final del cual se sitúa un botón "Select" que, tras pulsarlo, abrirá un dialogo "Graphs".

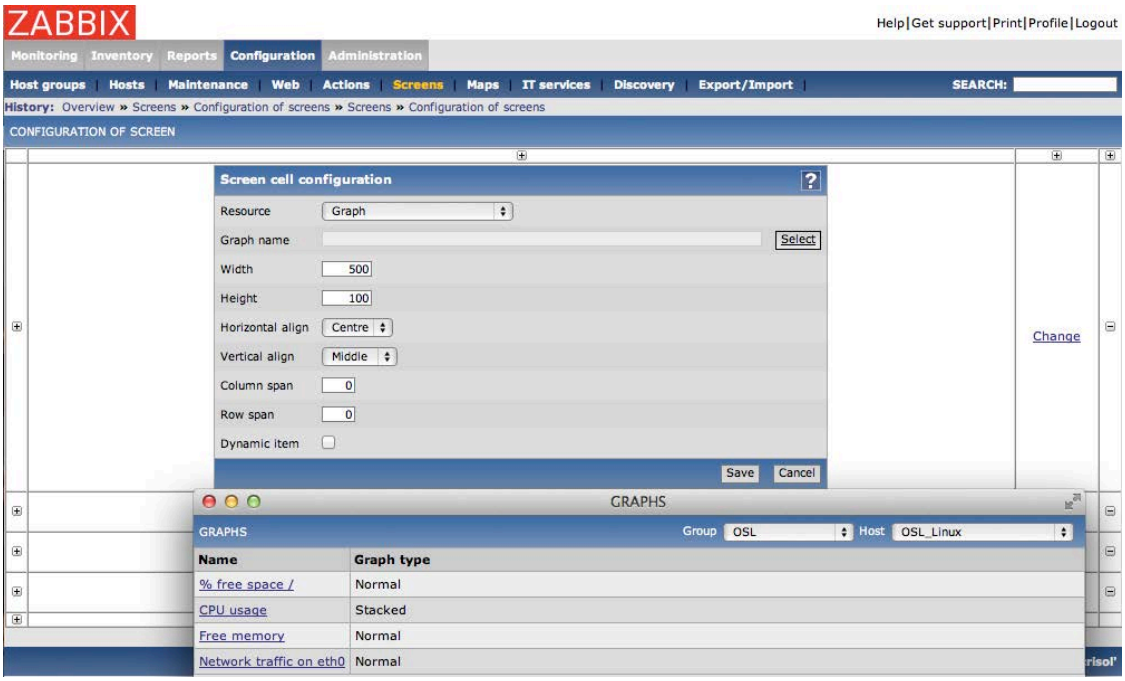

Ilustración 104 Pantalla de inserción de gráfica en screen

En el diálogo "Graphs" se presentan dos menús desplegables, el menú "Group", para elegir uno de los grupos de hosts definidos, y el menú "Host", en el que aparecen los hosts y plantillas definidas. Seleccionamos el grupo "OSL" y el host "osl-prox1", tras lo que aparecerá una tabla con las cuatro gráficas definidas en ese host. Se selecciona "CPU usage" y se pulsa sobre el botón "Save" para establecer la gráfica en el cuadrante.

Se repite la operación para cada uno de los cuadrantes de la columna de la izquierda, seleccionando, para cada una, cada uno de los gráficos que quedaban para el host osl-prox1. A continuación se repite la operación en la columna de la derecha pero fijando el menú desplegable al valor "osl-prox2". Tras terminar de añadir todas las gráficas, obtendremos una pantalla como la de *Ilustración 105 Pantalla de edición de screen 2*.

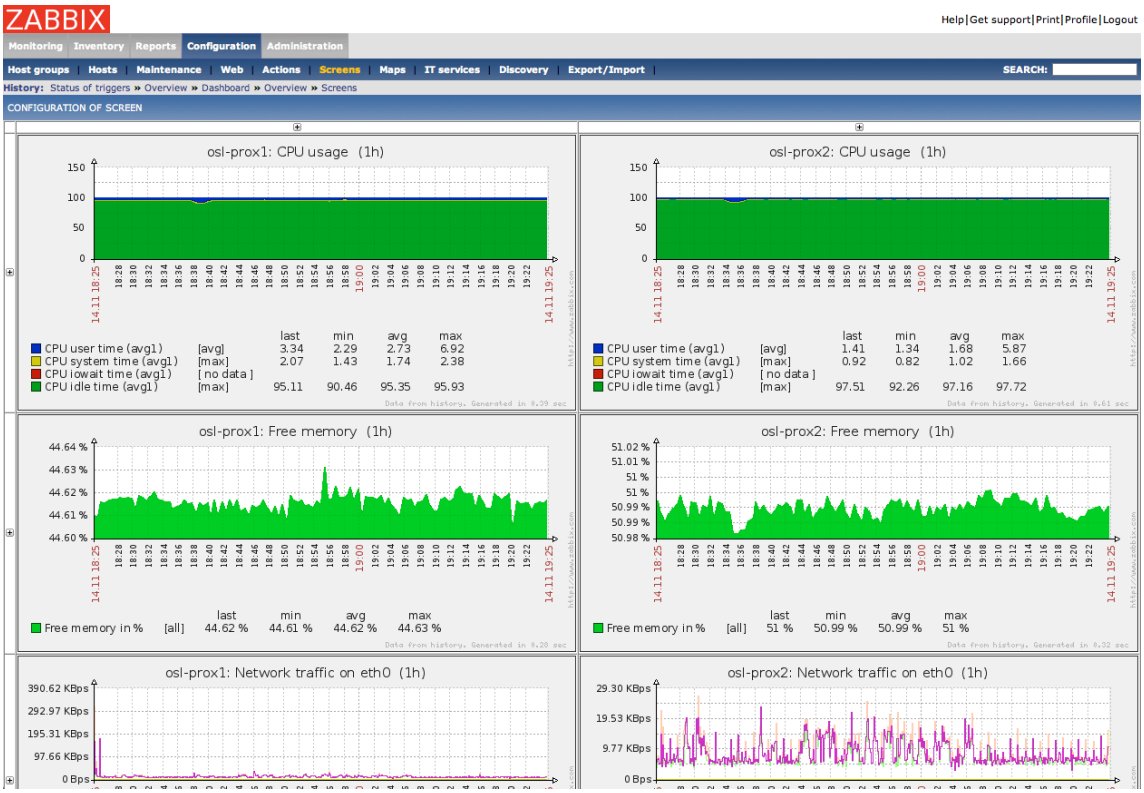

Ilustración 105 Pantalla de edición de screen 2

4.5.7.10. Definición de mapas

En los mapas se pueden disponer, de forma gráfica, varios hosts o grupos en una pantalla de la interfaz. El objetivo es poder visualizar cómodamente un grupo de hosts y si éstos presentan algún problema.

Para definir un mapa, se accede a la interfaz de Zabbix y se selecciona la pestaña “Configuration”, de entre los menús que se presentan en esta pestaña, se selecciona “Maps”, lo que abrirá la pantalla de mapas.

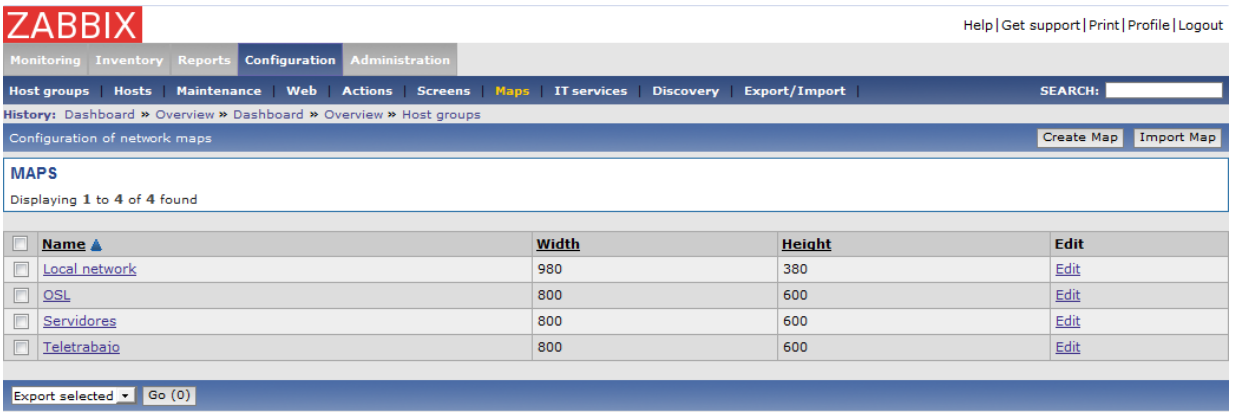

Ilustración 106 Pantalla de mapas

Para crear el nuevo mapa se pulsa el botón "Create Map" de la esquina superior derecha. Alternativamente, se puede definir el mapa localmente codificándolo en un archivo XML y subirlo al servidor mediante el botón "Import Map". Puesto que es muy costoso definir los mapas mediante XML, sólo se utilizará esta alternativa para importar mapas previamente exportados de un servidor de Zabbix (ya sea para restaurar un backup, realizar una migración o editar un mapa manualmente).

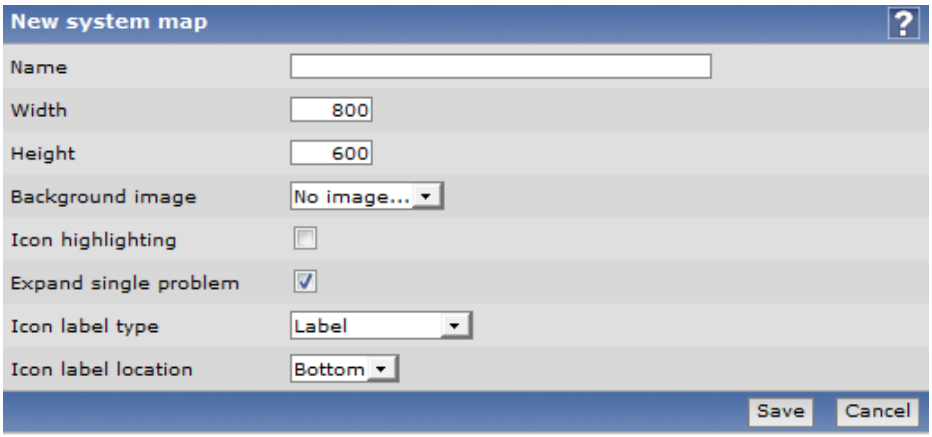

Ilustración 107 Pantalla de definición de mapa

Tras pulsar el botón "Create Map" se accederá a la pantalla de definición de mapa. En esta pantalla damos un nombre al mapa escribiendo en el campo "Name" "**Local network**", para indicar que se tratará de un mapa de la red monitorizada. Como, en principio, se desconoce el tamaño que tendrá el mapa, no se modifica ningún parámetro de los que vienen definidos por defecto, podrán ser editados más adelante si es necesario. Se pulsa el botón "Save" para terminar de definir el mapa, tras lo que se volverá a la pantalla de mapas en el que aparecerá el nuevo mapa. En la pantalla de mapas se pincha con el ratón sobre el nuevo mapa, "Local network", para acceder a la pantalla de edición del mapa, donde se pueda definir los elementos que contendrá.

En la esquina superior izquierda aparece una etiqueta "Element", junto a dos botones: "+" y "-". Para añadir un nuevo elemento al mapa, se pulsa el botón "+" para que se despliegue el diálogo de configuración de elemento.

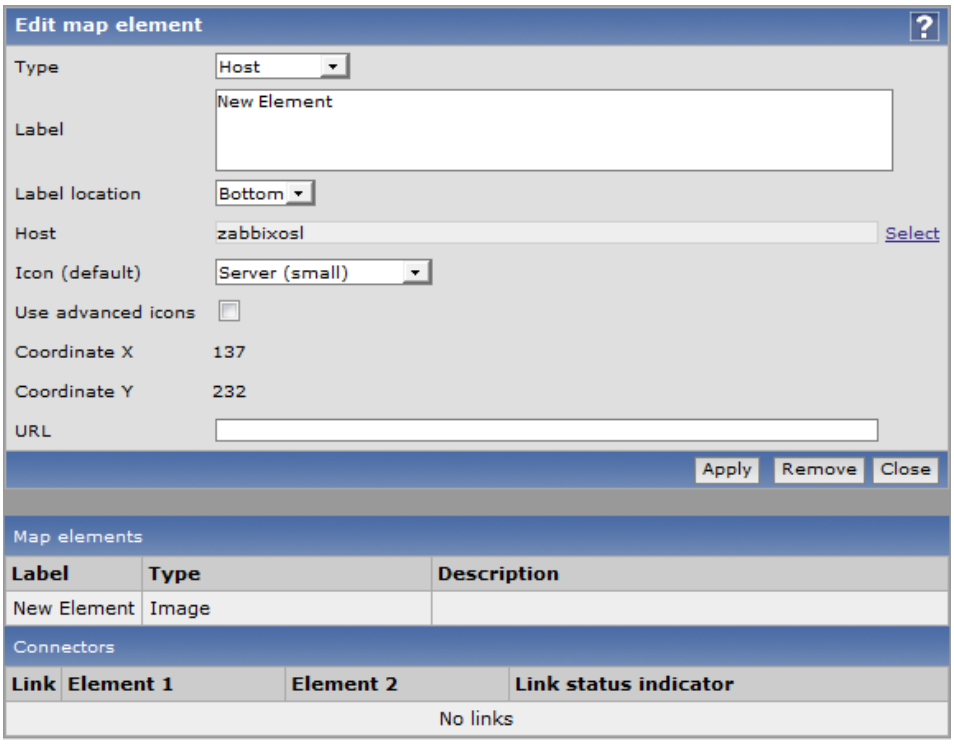

Ilustración 108 Diálogo de configuración de elemento

En primer lugar se añadirá un elemento que represente al nodo del clúster de Teletrabajo Prox1. Para ello, indicamos que se va a añadir un host seleccionando "**Host**" en el menú desplegable "Type". Para poder identificarlo, se ha de definir una etiqueta que aparecerá junto al icono del host en el mapa, como el equipo se llama Prox1, introducimos en el campo "Label" el mismo nombre "**prox1**".

En el campo "Label location" es posible indicar en qué posición, respecto al icono del host, queremos que aparezca la etiqueta (superior, inferior, izquierda o derecha). En este caso seleccionamos en el desplegable el valor "**Right**" para situar la etiqueta a la derecha.

Para que en el mapa se puedan mostrar avisos referentes a *trigger*s de hosts, hay que definir una asociación entre un host monitorizado y el elemento del mapa. La asociación del elemento con un host se hace en el campo "Host", que al final del mismo tiene un botón denominado "Select". Tras pulsar ese botón, se abrirá un nuevo diálogo con la lista de hosts disponibles para un determinado grupo. Se selecciona el grupo "OSL" y a continuación el host "osl-prox1". A partir de este momento, cada vez que se active un *trigger* en Prox1, aparecerá un mensaje en color rojo junto a la etiqueta del elemento con el nombre del *trigger* activado (o un número si hay más de un *trigger* activado).

Es posible cambiar la representación gráfica que tendrá el elemento dentro del mapa, para ello hay un menú desplegable llamado "Icon (default)", que permite

dibujar servidores, estaciones de trabajo, nodos de enrutamiento, etc. Puesto que Prox1 es un servidor, escogemos el icono "**Server**".

Finalmente, deseamos añadir una funcionalidad nueva: cuando pinchemos sobre el icono del elemento se muestrarán todas las gráficas definidas en ese host, de manera que desde el mapa tengamos acceso a toda la información más significativa de toda la red monitorizada. Para poder añadir esta funcionalidad, previamente es necesario haber creado una pantalla (o screen) de gráficas para cada host monitorizado y copiar la URL con la que se accede a esa pantalla, para, a continuación, pegarla en el campo "URL" del diálogo de configuración del elemento.

Para terminar, se pulsa sobre el botón "Apply", que almacena la configuración del nodo, apareciendo dibujado sobre el mapa el icono de un servidor etiquetado como "prox1". Este icono se puede arrastrar para colocarlo en la posición del mapa deseado.

Repetimos la operación otras tres veces para añadir los servidores correspondientes al segundo nodo del clúster de teletrabajo, el servidor de monitorización y el servidor anfitrión del servidor de monitorización.

Además de pueden definir elementos que representen otros mapas, de forma que se pueden crear sub-mapas accesibles pinchando sobre el elemento que los representa y ayudándonos a organizar más claramente el mapa. La definición de un elemento mapa se hace igual que la definición de un host en el diálogo de configuración del elemento, salvo en el campo "Type", se ha de seleccionar el valor "**Map**" y, en lugar de indicar un host vinculado en "Host", se define un mapa vinculado. Se crean mapas para agrupar todas las máquinas de Teletrabajo denominado "**teletrabajo**", otro para agrupar los servidores virtuales con los que se desea ampliar en el futuro el servicio de teletrabajo, que llamamos "**servidores**", y, por último, un mapa dentro del cual se situarán todos los equipos de la Oficina de Software Libre que se llamará "**OSL**".

Para completar el mapa se representan las conexiones de red, de una forma muy simplificada y esquemática, que existen entre los elementos. Esta representación de conexiones no tiene ninguna funcionalidad dentro del mapa, sólo sirve para aportar una visión esquemática de la disposición de los equipos de la red al administrador, para que así pueda localizarlos más rápidamente. Para dibujar una conexión entre dos elementos, primero se ha de seleccionar uno y, mientras se mantiene pulsada la tecla "Ctrl", seleccionar el elemento con el que se desea conectar, a continuación se pulsa el botón "+" de la esquina superior izquierda situado junto a la etiqueta "Link".

Una vez dibujadas las conexiones entre los elementos, pulsamos sobre el botón "Save" para terminar la edición del mapa y guardar los cambios.

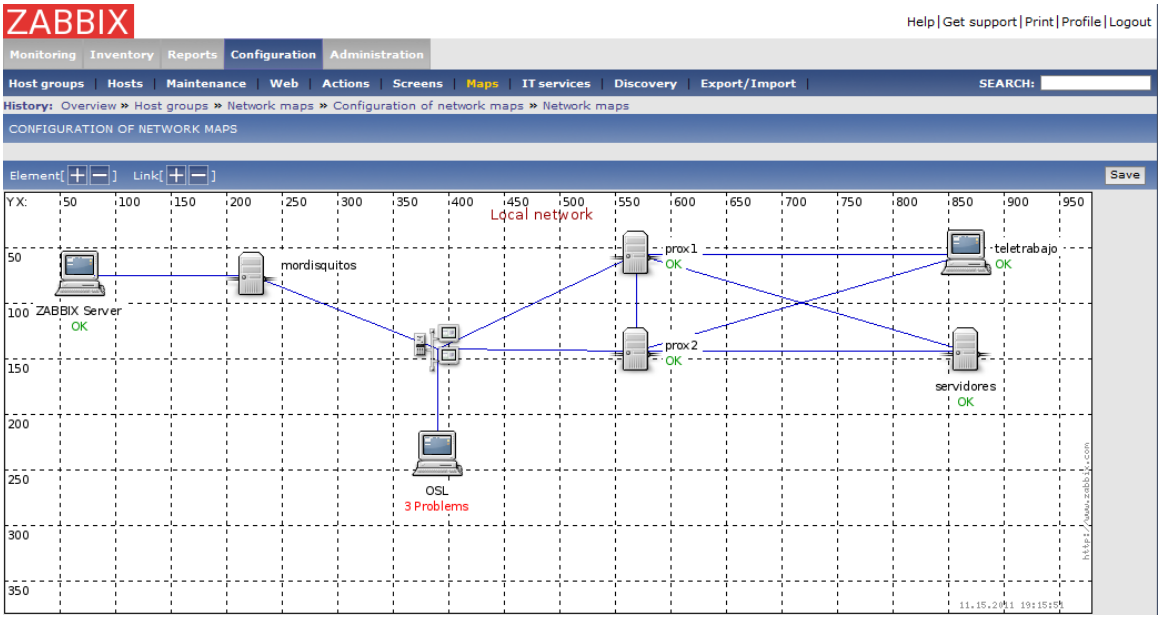

Ilustración 109 Pantalla de edición del mapa

4.6. Resumen del proyecto

Los siguientes apartados tienen como objetivo realizar una estimación del consumo de recursos invertidos en el desarrollo del proyecto, tanto económicos, como referentes a la duración del mismo.

En el primer apartado, se realiza un resumen del tiempo invertido en cada una de las fases del ciclo de vida por las que ha pasado el proyecto, desglosando cada una de las fases en las tareas que la componen. Esto permite realizar un cálculo del tiempo total que se le ha dedicado al proyecto, para poder realizar los cálculos de costes, incluidos en el segundo apartado.

El segundo apartado realizará un cálculo de los costes del proyecto, a partir de la inversión en tiempo, la infraestructura hardware y software utilizada.

4.6.1. Situación actual

A continuación se muestra el resumen con el tiempo que se ha invertido en cada fase y tarea del proyecto. Al no haber finalizado el servicio de Teletrabajo en el momento de la redacción de este documento, se ha estimado la duración de la fase de producción y mantenimiento hasta la primera semana de diciembre. El resto de fechas de inicio y final de cada tarea, se han calculado como valores aproximados basándose en ciertos hitos del proyecto y el tiempo que se les dedicó.

Hay que destacar que la fase de producción engloba todos los sistemas implicados entre proyecto predecesor y el actual, incluyendo el servicio de Teletrabajo y el de monitorización.

La siguiente tabla muestra las tareas agrupadas según las fases expuestas en este documento. Concretamente, los bloques análisis, diseño, codificación (o implantación), pruebas y mantenimiento (o producción).

ID	Nombre de tarea	Duración	Comienzo	Fin	Predecesora
1	**Análisis**	**110 días**	**mar 15/02/11**	**sáb 04/06/11**	**-**
2	Casos de uso	4 días	mar 15/02/11	vie 18/02/11	-
3	Requisitos software	6 días	sáb 19/02/11	jue 24/02/11	2
4	Reanálisis monitorización 1	1 día	mar 24/05/11	mar 24/05/11	24
5	Reanálisis monitorización 2	1 día	sáb 04/06/11	sáb 04/06/11	25
6	**Diseño**	**102 días**	**vie 25/02/11**	**lun 06/06/11**	**-**
7	Diseño arquitectónico	6,5 días	vie 25/02/11	jue 03/03/11	3
8	Diseño control de acceso	7 días	vie 04/03/11	jue 10/03/11	7
9	Diseño backup	21 días	vie 04/03/11	jue 24/03/11	7
10	Diseño monitorización	8 días	vie 04/03/11	vie 11/03/11	7
11	Rediseño monitorización 1	1 día	mié 25/05/11	mié 25/05/11	4
12	Rediseño monitorización 2	1 día	lun 06/06/11	lun 06/06/11	5
13	**Codificación/ Implantación**	**76 días**	**vie 25/03/11**	**mié 08/06/11**	**-**
14	Implantar control de acceso	7 días	vie 25/03/11	jue 31/03/11	8
15	Implantar backup	28 días	vie 25/03/11	jue 21/04/11	9
16	Despliegue servidor de monitorización	4 días	vie 25/03/11	lun 28/03/11	10
17	Despliegue agente de monitorización	2 días	mar 29/03/11	mié 30/03/11	16
18	Configurar servicio de monitorización	28 días	jue 31/03/11	mié 27/04/11	17

19	Reconfigurar servicio de monitorización 1	2 días	jue 26/05/11	vie 27/05/11	11
20	Reconfigurar servicio de monitorización 2	2 días	mar 07/06/11	mié 08/06/11	12
21	**Pruebas**	**49 días**	**jue 28/04/11**	**mié 15/06/11**	
22	Pruebas de verificación	9 días	jue 28/04/11	vie 06/05/11	15; 18; 14
23	Pruebas de rendimiento	9 días	sáb 07/05/11	dom 15/05/11	22
24	Pruebas de rendimiento 2	7 días	lun 16/05/11	dom 22/05/11	23
25	Pruebas de rendimiento 3	7 días	sáb 28/05/11	vie 03/06/11	19
26	Pruebas de rendimiento 4	7 días	jue 09/06/11	mié 15/06/11	20
27	**Producción/ Mantenimiento**	**199 días**	**lun 16/05/11**	**vie 02/12/11**	**23**
	Total	**289 días**	**mar 15/02/11**	**vie 02/12/11**	-

Tabla 140 Tabla de resumen de la duración de las fases del proyecto

El diagrama de Gantt que se presenta en *Ilustración 110 Diagrama de Gantt 1* e *Ilustración 111 Diagrama de Gantt 2*, muestra las tareas agrupadas según el flujo de trabajo realizado. Presentándose la primera iteración del ciclo de vida formado por las fases de análisis, diseño, codificación y pruebas, que engloban al control de acceso, backup y sistema de monitorización. En cambio, las dos siguientes iteraciones del ciclo de vida, únicamente se refieren al sistema de monitorización. Esto es así porque estas dos últimas iteraciones no pueden realizarse hasta que el servicio de Teletrabajo esté en fase de producción y, por tanto, se puedan realizar mediciones de uso reales para poder ajustar los umbrales del sistema de monitorización.

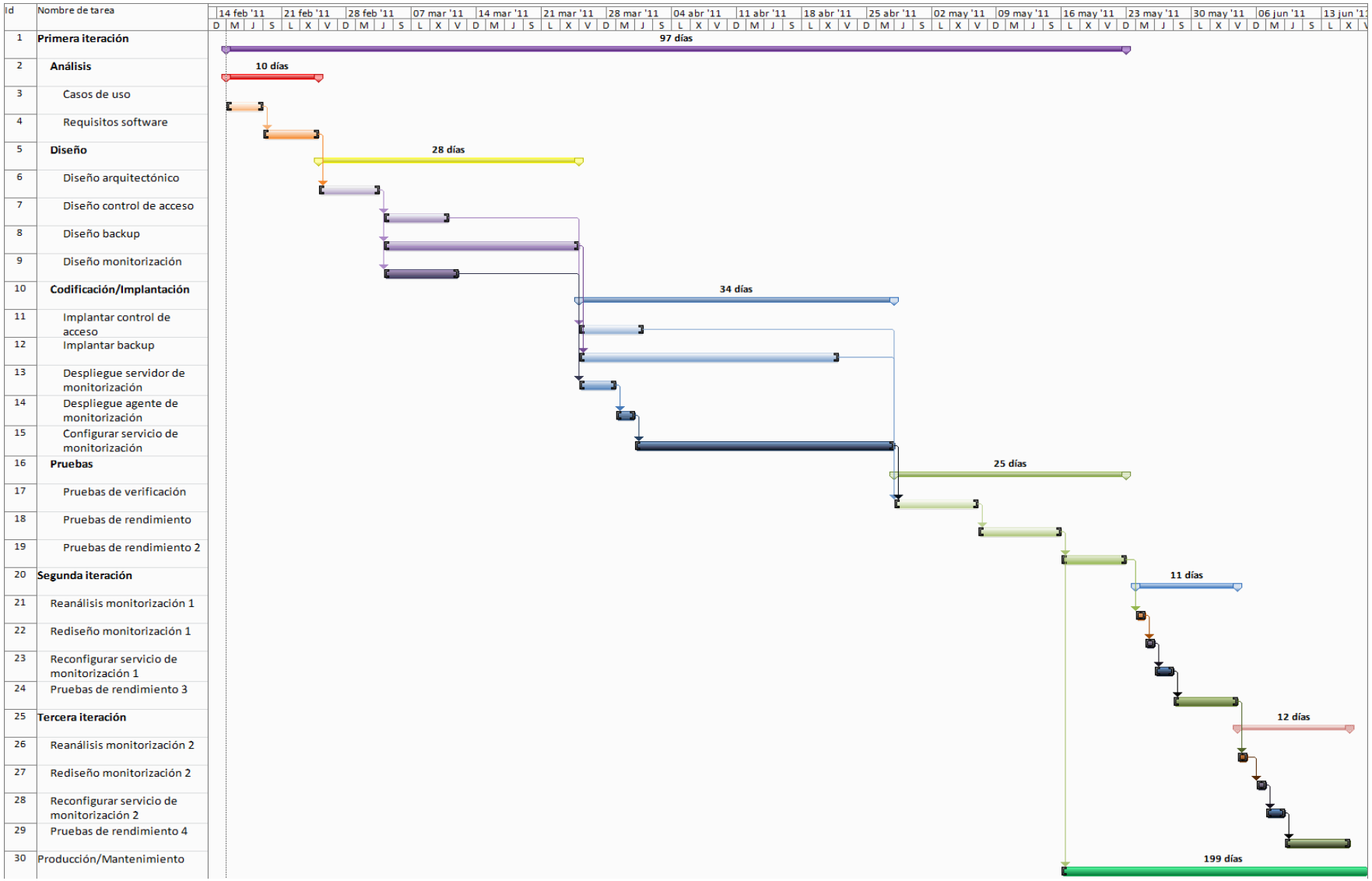

Ilustración 110 Diagrama de Gantt 1

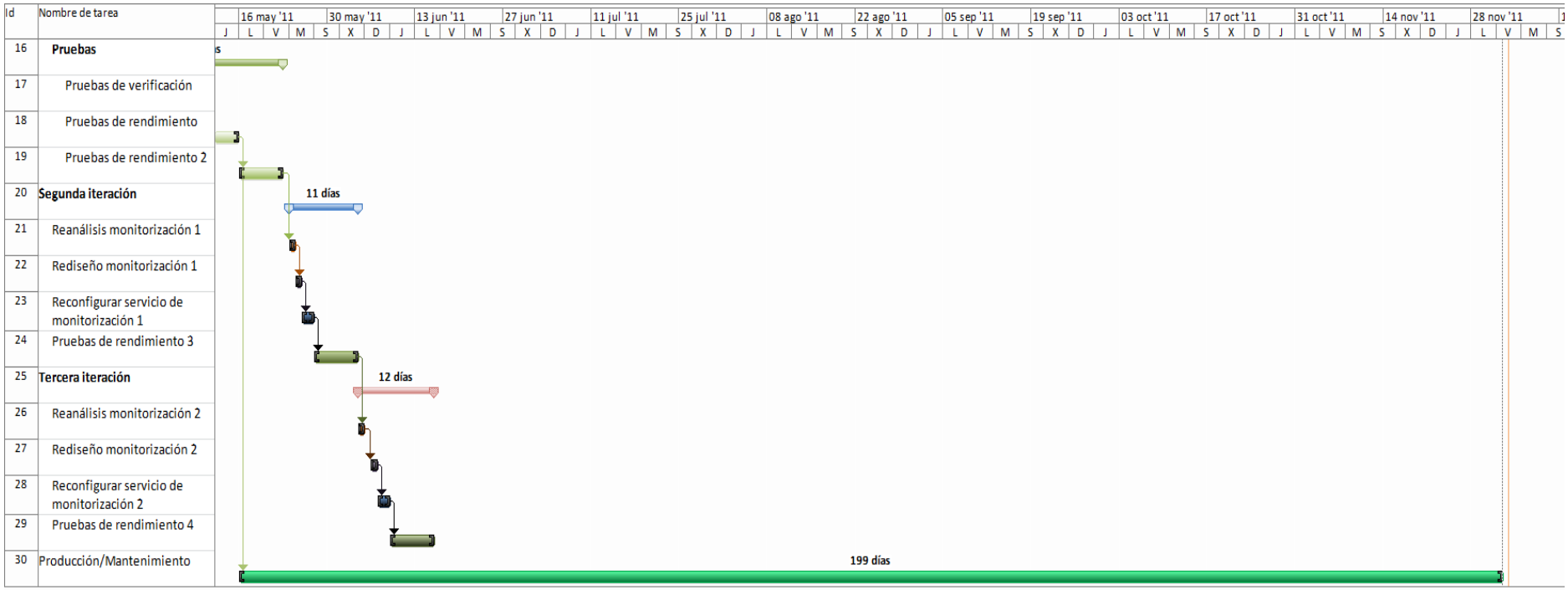

Ilustración 111 Diagrama de Gantt 2

4.6.2. Resumen de costes

El resumen del proyecto nos ha ofrecido una duración total de 289 días, correspondientes a 9,63 meses. Puesto que únicamente ha habido una persona dedicada al proyecto, esos 9.63 meses para una persona será la dedicación total de personal al proyecto. El personal encargado del desarrollo del proyecto es también el encargado del mantenimiento, motivo por el cual se incluye el tiempo de producción para el cálculo del presupuesto.

Dado que todo el software utilizado ha sido libre (con una excepción de software gratuito), y además no se han adquirido productos de asistencia por parte de las compañías responsables de ese software, no se ha originado ningún coste adicional relativo a software.

Con relación al hardware, se cuenta con que los servidores de Teletrabajo ya han sido amortizados en el proyecto anterior (al estar incluidos en su presupuesto). En cambio, en este proyecto se incluye el servidor de monitorización que, a pesar de ser una máquina virtual, se calculará cuál es su coste proporcional. Para calcular el porcentaje de uso del servidor de monitorización, nos basamos en la proporción de memoria RAM que utiliza el equipo con respecto al equipo anfitrión. Teniendo en cuenta que el servidor anfitrión dispone de 8 GB de memoria, y la máquina virtual de monitorización 3 GB, entonces la dedicación del servidor anfitrión a la monitorización será del 0,375.

1.- Autor:	
Alberto González Piedra	
2.- Departamento:	
Departamento de Informática	
3.- Descripción del proyecto:	
- Título	Sistema de seguridad y monitorización del servicio de Teletrabajo de la UC3M basado en software libre
- Duración	9,63 meses
Tasa de costes indirectos	20%
4.- Presupuesto total del Proyecto:	
	31.258 €
5.- Desglose presupuestario:	

PERSONAL				
Apellidos, Nombre	**Categoría**	**Dedicación**	**Coste mes**	**Coste**
González Piedra, Alberto	Ingeniero	9,63 meses	2.694,39 €	25.946,98€

EQUIPOS					
Descripción	**Coste**	**% Uso dedicado**	**Dedicación**	**Período de depreciación**	**Coste imputable**
Servidor: HP Proliant DL380 G5	6.017,65 €	37,5	9,63 meses	60	101,39 €

SUBCONTRATACIÓN DE TAREAS
No aplicable

OTROS COSTES DIRECTOS DEL PROYECTO
No aplicable.

6.- Resumen de costes:	
Detalle	**Costes totales**
Personal	25.947 €
Amortización	101 €
Subcontratación de tareas	0 €
Costes de funcionamiento	0 €
Costes indirectos	5.210 €
Total	31.258 €

Tabla 141 Presupuesto del proyecto

4.7. Mantenimiento del proyecto

La fase de producción del sistema de Teletrabajo tiene una duración de seis meses, como se detalla en el proyecto anterior(Gil Bázquez, 2011). Este periodo de tiempo se ha de dedicar al mantenimiento del sistema de Teletrabajo, apoyándose para ello en las herramientas desarrollas en proyecto actual. En concreto, las tareas de mantenimiento tendrán como eje central el sistema de monitorización, que será el encargado de supervisar el correcto funcionamiento del servicio de Teletrabajo y notificar al administrador ante cualquier problema, siendo así el servidor de Zabbix el iniciador de la mayoría de tareas de mantenimiento.

Otra parte importante del mantenimiento del proyecto es obtener estadísticas sobre el uso de las máquinas virtuales por parte de los teletrabajadores, de manera que puedan ser ajustadas, en caso de detectar alguna deficiencia, y poder mejorar el servicio ofrecido al teletrabajador. Para llevar a cabo esta función, son imprescindibles las gráficas definidas en el servidor de monitorización, que permiten detectar tendencias de uso y consumo de recursos de las máquinas virtuales.

A continuación se detallará el conjunto de procedimientos a seguir ante las incidencias que puedan surgir que interrumpan el servicio de Teletrabajo. Estos procedimientos se recogerán en lo que se conoce como plan de contingencia del servicio de Teletrabajo:

Identificador: PC-01	
Nombre	Levantar máquina virtual.
Incidencia	Máquina virtual de teletrabajo caída.
Gravedad	Media.
Responsables	Administrador de Teletrabajo.
Procedimiento	1. Se recibe un correo de notificación con el asunto y mensaje “Server tt-<nombre_usuario> is unreachable the last 8 min: PROBLEM”. 2. Se accede al servidor de monitorización zabbixosl.uc3m.es. 3. Se selecciona la pestaña “Monitoring” en Zabbix. 4. Se selecciona el menú “Overview” en “Monitoring”. 5. Se selecciona “Teletrabajo” en el menú desplegable “Group”. 6. Se comprueba el color del *trigger* “Server {HOSTNAME} is unreachable the last 8 min” a. Si el recuadro del *trigger* está en color verde, el equipo se ha recuperado y se finaliza. b. Si el recuadro del *trigger* está en rojo, persiste el problema y se continúa en el paso 7. 7. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 8. Se selecciona “Máquinas virtuales” en el menú de la izquierda.

Identificador: PC-01	
	9. Se selecciona la máquina virtual tt-<nombre_usuario>. 10. Se selecciona "Open VNC console" para abrir la consola de la máquina. a. Si la máquina virtual responde se espera 5 minutos, si a los 5 minutos se desactiva el *trigger*, la máquina se ha recuperado y se finaliza. Si no se desactiva el *trigger* se reinicia la máquina virtual. b. Si la máquina virtual no responde se resetea pulsando sobre el botón "Restablecer" y se finaliza. 11. Si tras reiniciar la máquina virtual sigue sin funcionar se salta a PC-02.

Tabla 142 Plan de contingencia PC-01

Identificador: PC-02	
Nombre	Restaurar máquina virtual.
Incidencia	Máquina virtual de teletrabajo no funciona.
Gravedad	Media.
Responsables	Administrador de Teletrabajo.
Procedimiento	1. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 2. Se selecciona "Máquinas virtuales" en el menú de la izquierda. 3. Se verifica en que nodo (N) del clúster de teletrabajo se localiza la máquina afectada y cuál es su identificador. 4. Se accede con una sesión remota *ssh* al servidor de teletrabajo que hospeda la máquina virtual: prox<N>. 5. Se ejecuta con privilegios de súper-usuario el comando "/usr/local/bin/vmtools/vmbackup –r <identificador_maquina>", siendo <identificador_maquina> el identificador de la máquina virtual afecta. Se mostrará una lista de

Identificador: PC-02	
	backups de la que <nombre_backup> es el backup más moderno que no se ha probado. Si no queda ningún backup se salta a PC-03. 6. Se restaura el backup ejecutando el comando ““/usr/local/bin/vmtools/vmbackup –r <identificador_maquina> <nombre_backup>” y se espera hasta que finalice la ejecución del comando. 7. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 8. Se selecciona “Máquinas virtuales” en el menú de la izquierda. 9. Se selecciona la máquina virtual con identificador <identificador_maquina>. 10. Se selecciona “Open VNC console” para abrir la consola de la máquina. 11. Si la máquina no funciona, volver al paso 5. Si la máquina funciona se finaliza.

Tabla 143 Plan de contingencia PC-02

Identificador: PC-03	
Nombre	Restaurar máquina virtual con backup corporativo.
Incidencia	Máquina virtual de teletrabajo caída tras restaurar los backups locales.
Gravedad	Media-alta.
Responsables	Administrador de Teletrabajo.
Procedimiento	1. Se solicita al servicio de backup corporativo el backup realizado una semana antes al último backup restaurado. 2. Se accede al servidor de Teletrabajo que hospeda la máquina virtual afectada. 3. Se elimina la máquina virtual afectada mediante el comando “qe destroy <identificador_maquina>. 4. Se restaura el fichero obtenido del backup corporativo <backup_corporativo> con el comando

Identificador: PC-03	
	"qmrestore <backup_corporativo> <identificador_maquina>. 5. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 6. Se selecciona "Máquinas virtuales" en el menú de la izquierda. 7. Se selecciona la máquina virtual con identificador <identificador_maquina>. 8. Se selecciona "Open VNC console" para abrir la consola de la máquina. 9. Si la máquina no funciona, volver al paso 1. Si la máquina funciona se finaliza.

Tabla 144 Plan de contingencia PC-03

Identificador: PC-04	
Nombre	Levantar escritorio remoto.
Incidencia	Servicio de escritorio remoto RDP está inactivo en una máquina virtual.
Gravedad	Media.
Responsables	Administrador de máquinas virtuales.
Procedimiento	1. Se recibe un correo de notificación con el asunto y mensaje "RDP listening on tt-<nombre_usuario>: PROBLEM". 2. Se accede al servidor de monitorización zabbixosl.uc3m.es. 3. Se selecciona la pestaña "Monitoring" en Zabbix. 4. Se selecciona el menú "Overview" en "Monitoring". 5. Se selecciona "Teletrabajo" en el menú desplegable "Group". 6. Se comprueba el color del *trigger* "RDP listening on {HOSTNAME}" a. Si el recuadro del *trigger* está en color verde, el

Identificador: PC-04	
	servicio se ha recuperado y se finaliza. b. Si el recuadro del *trigger* está en rojo, persiste el problema y se continúa en el paso 7. 7. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 8. Se selecciona “Máquinas virtuales” en el menú de la izquierda. 9. Se selecciona la máquina virtual tt-<nombre_usuario>. 10. Se selecciona “Open VNC console” para abrir la consola de la máquina. 11. Se inicia sesión en la máquina virtual con una cuenta de administrador. 12. Se abre el menú contextual del icono “Mi PC” y se selecciona “Propiedades”. 13. Se selecciona la pestaña “remoto”. 14. Se activa la casilla “Permitir que los usuarios se conecten de manera remota a este equipo”.

Tabla 145 Plan de contingencia PC-04

Identificador: PC-05	
Nombre	Levantar firewall.
Incidencia	Firewall de un servidor de Teletrabajo inactivo.
Gravedad	Baja-Media.
Responsables	Administrador del servidor de Teletrabajo.
Procedimiento	1. Se recibe un correo de notificación con el asunto y mensaje “Firewall down on <nombre_servidor>: PROBLEM”. 2. Se accede al servidor de monitorización zabbixosl.uc3m.es. 3. Se selecciona la pestaña “Monitoring” en Zabbix. 4. Se selecciona el menú “Overview” en “Monitoring”. 5. Se selecciona “Teletrabajo” en el menú desplegable

Identificador: PC-05	
	"Group". 6. Se comprueba el color del *trigger* "Firewall down on {HOSTNAME}" a. Si el recuadro del *trigger* está en color verde, el equipo se ha recuperado y se finaliza. b. Si el recuadro del *trigger* está en rojo, persiste el problema y se continúa en el paso 7. 7. Se accede mediante una sesión remota *ssh* al servidor de Teletrabajo <nombre_servidor>. 8. Se ejecuta el comando "shorewall start" con privilegios de súper-usuario.

Tabla 146 Plan de contingencia PC-05

Identificador: PC-06	
Nombre	Persistencia de servicio ante caída en un servidor.
Incidencia	Caída de un nodo del servicio de Teletrabajo.
Gravedad	Alta.
Responsables	Administrador del servidor de Teletrabajo.
Procedimiento	1. Se accede mediante una sesión remota *ssh* al nodo en funcionamiento del servicio de Teletrabajo. 2. Se ejecuta el comando "ls –l /etc/qemu-server/prox<N>", siendo N el número de nodo del clúster caído. Se obtiene el nombre del último directorio modificado <directorio>. 3. Se ejecuta el comando "cp /etc/qemu-server/prox<N>/<directorio>/* /etc/qemu-server/. 4. Se ejecuta el comando "/usr/local/bin/vmtools/vmmanagement start 101 124" con privilegios de súper-usuario.

Tabla 147 Plan de contingencia PC-06

Identificador: PC-06	
Nombre	Persistencia de servicio ante parada de un servidor para

Identificador: PC-06	
	mantenimiento.
Incidencia	Pérdida de servicio de Teletrabajo en las máquinas hospedadas por el servidor parado.
Gravedad	Baja.
Responsables	Administrador de Teletrabajo, administrdor del servidor de Teletrabajo.
Procedimiento	1. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 2. Se selecciona “Máquinas virtuales” en el menú de la izquierda. 3. Se selecciona la flecha roja junto a la primera máquina virtual del nodo que se desea parar, y se selecciona “Emigrar”. 4. Se marca la casilla “Emigración en linea”. 5. Se selecciona “migrate” y se espera que finalice la migración. 6. Si quedan máquinas virtuales en el nodo del servicio de Teletrabajo que se desea parar, se vuelve al paso 3. Si no quedan máquinas virtuales en el nodo a parar se salta a 7. 7. Se realiza la tarea de mantenimiento o reparación en el nodo sin máquinas del servicio de Teletrabajo. 8. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 9. Se selecciona “Máquinas virtuales” en el menú de la izquierda. 10. Si hay una máquina hospedada en nodo del servicio de Teletrabajo al que no pertenece, se selecciona la flecha roja junto a la primera máquina virtual del nodo que se desea parar, y se selecciona “Emigrar”. Si todas las máquinas están en su nodo correspondiente se finaliza. 11. Se marca la casilla “Emigración en linea”. 12. Se selecciona “migrate” y se espera que finalice la migración. Se vuelve al paso 10.

Tabla 148 Plan de contingencia PC-07

Capítulo V

Plan de pruebas

Este capítulo consta de dos partes. En la primera, que se llevará a cabo tras la fase de implantación, se detallará el plan de pruebas que se realizará para verificar que se ha implementado toda la funcionalidad descrita durante el diseño.

La segunda parte se inicia al final de la fase de implantación y continuará una vez comenzada la fase de producción. Esta parte es la referente a las pruebas de rendimiento. Las pruebas de rendimiento constan de dos partes: una primera, anterior a la fase de producción, para asegurar que el servicio de Teletrabajo se comportará correctamente cuando lo usen los teletrabajadores; y una segunda parte, consistente en la medición del uso de los equipos y optimización de los mismos, para mejorar el servicio.

5.1. Pruebas de verificación

El objetivo de este punto es verificar que el sistema cumple con las especificaciones, mediante los resultados de los procedimientos que se realizan. Es decir, se comprobará que toda la funcionalidad y todas las características, especificadas en los requisitos software del proyecto, han sido implementadas.

A continuación se muestran las tablas con los procedimientos que se han de seguir para verificar los requisitos:

Identificador: PV-01	
Título:	Realización de un backup.

Identificador: PV-01	
Descripción:	Comprobar que se puede generar un backup de una máquina virtual y restaurarlo.
Requisitos relacionados:	RSF-32, RSF-33, RSF-36, RSF-37, RSF-38
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 2. Ejecutar el comando "/usr/local/bin/vmtools/vmbackup" para que se muestren las opciones del comando. 3. Ejecutar el comando para que mantenga una copia de la máquina virtual y se eliminen copias obsoletas tras un día de la máquina virtual 101 (rango de una máquina): "/usr/local/bin/vmtools/vmbackup -c 1 1 101 101". 4. Acceder a la interfaz del servicio de Teletrabajo mediante la dirección "prox1.uc3m.es". 5. Acceder al menú máquinas virtuales. 6. Seleccionar la flecha junto a la máquina virtual 101 y escoger la opción "Detener". 7. Seleccionar la flecha junto a la máquina virtual 101 y escoger la opción "Remove". 8. Volver a la sesión remota de ssh y ejecutar el comando "/usr/local/bin/vmtools –r 101" para que se muestren los backups disponibles de la máquina 101. 9. Ejecutar el comando "/usr/local/bin/vmtools/vmbackup –r 101 <nombre_backup>", siendo <nombre_backup> el nombre del fichero mostrado en el paso 8. 10. Volver a la interfaz del servicio de Teletrabajo. 11. Seleccionar la flecha junto a la máquina virtual 101 y escoger la opción "Inicio". 12. Seleccionar la máquina virtual 101. 13. Seleccionar "Open VNC console". 14. Comprobar cómo funciona la máquina virtual.
Criterio de	Se ha restaurado una máquina virtual desaparecida a

Identificador: PV-01	
aceptación:	partir de un backup.

Tabla 149: Prueba de verificación PV-01

Identificador: PV-02	
Título:	Realización de un backup en un rango de máquinas virtuales.
Descripción:	Comprobar que se puede generar el backup de cada una de las máquinas virtuales de un rango.
Requisitos relacionados:	RSF-34
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 2. Tomar la hora y fecha actual ejecutando el comando "date". 3. Ejecutar el comando para que mantenga una copia de cada máquina virtual y se eliminen copias obsoletas tras un día de las máquina virtuales comprendidas entre la 101 y la 105: "/usr/local/bin/vmtools/vmbackup -c 1 1 101 105". 4. Ejecutar el comando "ls –Rl /var/lib/vz/snapshot". para comprobar que hay un backup (fichero *tgz*) para las máquinas virtuales 101, 102, 103, 104 y 105 con fecha y hora superior a la tomada en el paso 2.
Criterio de aceptación:	Comprobar que hay un backup (fichero *tgz*) para las máquinas virtuales 101, 102, 103, 104 y 105 con fecha y hora superior a la tomada en el paso 2.

Tabla 150: Prueba de verificación PV-02

Identificador: PV-03	
Título:	Realización de un backup cruzado.
Descripción:	Comprobar que se puede generar un backup cruzado de todas las máquinas de un nodo de Teletrabajo.
Requisitos	RSF-35, RSF-39, RSF-40, RSNF-10, RSNF-11, RSNF-

Identificador: PV-03	
relacionados:	20
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 2. Tomar la hora y fecha actual ejecutando el comando "date". 3. Ejecutar el comando "/usr/local/bin/vmtools/crossoverbackup 30" para decir que no borre los backups cruzados hechos en menos de 30 días. 4. Ejecutar el comando "ls /etc/qemu-server/prox2/" para que se muestre una carpeta (<carpeta_backup>) con la fecha tomada en el paso 2. 5. Ejecutar el comando "ls /etc/qemu-server/prox2/<carpeta_backup>" para visualizar todas las máquinas de las que se ha hecho backup cruzado. 6. Ejecutar el comando "cp /etc/qemu-server/prox2/<carpeta_backup>/* /etc/qemu-server/" para restaurar los backups. 7. Acceder al servidor de Teletrabajo prox2 mediante una sesión remota con *ssh*. 8. Ejecutar el comando "/usr/local/bin/vmtools/vmmanagement stop 118 124" para detener las máquinas. 9. Volver a la sesión remota de prox1 y ejecutar el comando "/usr/local/bin/vmtools/vmmanagement start 118 124". 10. Acceder a la interfaz del servicio de Teletrabajo mediante la dirección "prox1.uc3m.es". 11. Acceder al menú máquinas virtuales. 12. Seleccionar cada máquina del rango [118, 124] de prox1 y seleccionar "Open VNC console".
Criterio de aceptación:	Comprobar que todas las máquinas virtuales del rango [118, 124] funcionan correctamente.

Tabla 151: Prueba de verificación PV-03

Identificador: PV-04	
Título:	Realización de un backup corporativo.
Descripción:	Comprobar que se realiza correctamente el backup corporativo.
Requisitos relacionados:	RSF-41, RSNF-18, RSNF-22, RSNF-23
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh* un domingo antes de las 21:00. 2. Tomar la hora y fecha actual ejecutando el comando "date". 3. Ejecutar el comando "ls –Rl /var/lib/vz/backup" y comprobar que todos los ficheros tienen fecha de modificación de hace una semana. 4. Esperar a al menos las 2:00 del lunes siguiente. 5. Ejecutar el comando "ls –Rl /var/lib/vz/backup" y comprobar que todos los ficheros de backup (*.tgz) tienen una fecha de modificación superior a la tomada en el paso 2. 6. Comprobar la bandeja de correo electrónico.
Criterio de aceptación:	Se han creado nuevos ficheros de backup en el directorio /var/lib/vz/backup/ y se ha recibido un correo electrónico del servicio corporativo con los datos copiados.

Tabla 152: Prueba de verificación PV-04

Identificador: PV-05	
Título:	Consultar histórico de un *item*.
Descripción:	Se comprueba que el sistema de monitorización muestra el estado de las alertas y el valor de los elementos monitorizados.
Requisitos relacionados:	RSF-42, RSF-43, RSF-45, RSF-46, RSNF-25
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse.

Identificador: PV-05	
	2. Seleccionar la pestaña “Monitoring” y a continuación “Overview” de la interfaz para que se muestre el estado de todos los *trigger*s de un grupo. 3. En el menú desplegable “Group” se selecciona la opción “All”. 4. En el menú desplegable “Type” seleccionar la opción “Data” para mostrar el valor actual de los *item*s de los hosts de ese grupo. 5. Seleccionar cualquier valor mostrado de la fila denominada “Checksum of c:\autoexec.bat” y a continuación la opción “500 latest values”.
Criterio de aceptación:	Se muestra una tabla con el histórico de valores que ha tomado un *item*.

Tabla 153: Prueba de verificación PV-05

Identificador: PV-06	
Título:	Consultar el histórico de un *item* gráficamente.
Descripción:	Se comprueba que se puede consultar los últimos valores de un elemento monitorizado en una gráfica.
Requisitos relacionados:	RSF-44
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Monitoring” y a continuación “Overview” de la interfaz para que se muestre el estado de todos los *trigger*s de un grupo. 3. En el menú desplegable “Type” seleccionar la opción “Data” para mostrar el valor actual de los *item*s de los hosts de ese grupo. 4. Seleccionar cualquier valor mostrado y a continuación la opción “Last month graph”.
Criterio de aceptación:	Se muestra una gráfica con los valores del *item* seleccionado en función del tiempo durante un mes.

Tabla 154: Prueba de verificación PV-06

Identificador: PV-07	
Título:	Consultar esquema de la red.
Descripción:	Se comprueba que se puede visualizar una representación gráfica de la red monitorizada con los valores de los *trigger*s más importantes.
Requisitos relacionados:	RSF-47. RSF-48
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña "Monitoring" y a continuación "Maps" de la interfaz para que se muestre un esquema de la red. 3. Seleccionar "Local network" en el menú desplegable situado en la esquina superior derecha.
Criterio de aceptación:	Se ha restaurado un esquema de la red monitorizada con nodos y grupos de nodos, bajo los nombres de los cuales se indica el estado de los *trigger*s de ese nodo.

Tabla 155: Prueba de verificación PV-07

Identificador: PV-08	
Título:	Añadir un equipo a monitorizar.
Descripción:	Comprobar que se puede añadir y configurar un nuevo equipo para ser monitorizado.
Requisitos relacionados:	RSF-49, RSF-54, RSF-75, RSNF-13, RSNF-16, RSNF-17, RSNF-31
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 2. Ejecutar el comando "cat /etc/zabbix/zabbix_agentd.conf" y comprobar que la línea "Server" tiene el valor "zabbixosl.uc3m.es". 3. Ejecutar el comando "/etc/init.d/zabbix_agent start" para iniciar el agente. 4. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección

Identificador: PV-08	
	zabbixosl.uc3m.es e identificarse. 5. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 6. Seleccionar “Create host”. 7. Introducir “osl-prox1” en “name”, “Group nuevo” en “New group”, “prox1.uc3m.es” en “DNS name”, “DNS name” en “connect to”. 8. Seleccionar “Add” en “Linked templates”. 9. Seleccionar “OSL” en “Group” y “OSL_Linux_Server”. 10. Seleccionar “Save”.
Criterio de aceptación:	Aparece la pantalla de hosts del grupo con un nuevo elemento “osl-prox1” con el valor “Monitored” en la columna “Status”.

Tabla 156: Prueba de verificación PV-08

Identificador: PV-09	
Título:	Añadir elemento a monitorizar.
Descripción:	Comprobar que se puede añadir un nuevo *item* a un host o plantilla.
Requisitos relacionados:	RSF-50, RSF-52, RSF-53, RSNF-29
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 3. Seleccionar “Templates” en la esquina superior derecha de la pantalla. 4. Seleccionar “Items” en la fila de “OSL_Linux”. 5. Se selecciona “Create item”. 6. Introducir “SSH prueba” en “name”, “proc.num[sshd]” en key, “Numeric (unsigned)” en “Type of information”, “Decimal” en “Data type”. 7. Establecer “Active en status”.

Identificador: PV-09	
	8. Seleccionar “Save”. 9. Seleccionar la pestaña “Monitoring” y a continuación “Overview”. 10. Seleccionar “OSL” en el menú desplegable “Group”. 11. Seleccionar “Data” en “Type”.
Criterio de aceptación:	Aparece una fila llamada “SSH prueba” que tiene un valor en la columna osl-prox1 y otros equipos.

Tabla 157: Prueba de verificación PV-09

Identificador: PV-10	
Título:	Desactivar un elemento monitorizado.
Descripción:	Comprobar que se puede desactivar un *item* para que deje de ser monitorizado.
Requisitos relacionados:	RSF-51
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-09. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 4. Seleccionar “Hosts” en la esquina superior derecha de la pantalla. 5. Seleccionar “Items” en la fila de “osl-prox1”. 6. Buscar el *item* “SSH prueba” y seleccionar el enlace “Active” localizado en su fila. 7. Comprobar que el *item* aparece con el valor “Disabled” en la tabla. 8. Seleccionar la pestaña “Monitoring” y a continuación “Overview”. 9. Seleccionar “OSL” en el menú desplegable “Group”. 10. Seleccionar “Data” en “Type”.
Criterio de	En la fila denominada “SSH prueba” ha desaparecido el

Identificador: PV-10	
aceptación:	valor que tenía en la columna “osl.prox1”.

Tabla 158: Prueba de verificación PV-10

Identificador: PV-11	
Título:	Añadir un grupo a un host.
Descripción:	Comprobar que se puede asignar un host o plantilla a un grupo adicional.
Requisitos relacionados:	RSF-56
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 3. Seleccionar el nombre de cualquier host o plantilla. 4. Seleccionar un grupo de la lista “Other groups”. 5. Seleccionar el botón “<<”. 6. Seleccionar el botón “Save”. 7. Seleccionar la pestaña “Configuration” y a continuación “Host groups”.
Criterio de aceptación:	Comprobar que el host o plantilla modificado, aparece en la fila de la tabla correspondiente al grupo añadido.

Tabla 159: Prueba de verificación PV-11

Identificador: PV-12	
Título:	Quitar un host de un grupo.
Descripción:	Comprobar que se puede excluir un host o plantilla del grupo al que pertenecía.
Requisitos relacionados:	RSF-57
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse.

Identificador: PV-12	
	2. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 3. Seleccionar el nombre de cualquier host o plantilla. 4. Seleccionar un grupo de la lista “In groups”. 5. Seleccionar el botón “>>”. 6. Seleccionar el botón “Save”. 7. Seleccionar la pestaña “Configuration” y a continuación “Host groups”.
Criterio de aceptación:	Comprobar que el host o plantilla modificado, ha desaparecido de la fila de la tabla correspondiente al grupo añadido.

Tabla 160: Prueba de verificación PV-12

Identificador: PV-13	
Título:	Eliminación de un grupo.
Descripción:	Se comprueba que se pueden eliminar grupos definidos en el sistema.
Requisitos relacionados:	RSF-58
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 3. Seleccionar la casilla junto al nombre del grupo que se desea eliminar. 4. Seleccionar “Delete selected groups” del menú desplegable de la parte inferior de la pantalla. 5. Seleccionar “Go”.
Criterio de aceptación:	Comprobar que ha desaparecido el grupo en la tabla que se muestra.

Tabla 161: Prueba de verificación PV-13

Identificador: PV-14	
Título:	Eliminación de un elemento monitorizado.
Descripción:	Comprobar que se puede eliminar un *item* de un host para que deje de existir en el sistema.
Requisitos relacionados:	RSF-59
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-09. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña "Configuration" y a continuación "Hosts" de la interfaz. 4. Seleccionar "Templates" en la esquina superior derecha de la pantalla. 5. Seleccionar "Items" en la fila de "OSL_Linux". 6. Buscar el *item* "SSH prueba" y seleccionar la casilla que aparece junto a él. 7. Seleccionar "Delete selected" del menú desplegable de la parte inferior. 8. Seleccionar el botón "Go".
Criterio de aceptación:	Comprobar que el *item* "SSH prueba" ha desaparecido de la lista de *items* de la plantilla "OSL_Linux".

Tabla 162: Prueba de verificación PV-14

Identificador: PV-15	
Título:	Creación de una alerta.
Descripción:	Comprobar que se puede crear un *trigger* que supervise el valor de un *item*.
Requisitos relacionados:	RSF-60, RSF-63, RSF-71, RSNF-30
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-09. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse.

Identificador: PV-15	
	3. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 4. Seleccionar “Templates” en la esquina superior derecha de la pantalla. 5. Seleccionar “OSL” en el menú desplegable “Group” 6. Seleccionar “Triggers” en la fila de “OSL_Linux”. 7. Se selecciona “Create trigger”. 8. Establecer “SSH caído” en “Name” y “{OSL_Linux:net.tcp.service[ssh].last(0)}=0” en “Expression”. 9. Establecer “High” en el menú desplegable “Severity”. 10. Seleccionar la pestaña “Monitoring” y a continuación “Overview”. 11. Seleccionar “OSL” en el menú desplegable “Group”. 12. Seleccionar “Trigger” en “Type”.
Criterio de aceptación:	Comprobar que existe una fila “SSH caído” con un cuadro color verde o rojo en la columna “osl-prox1”.

Tabla 163: Prueba de verificación PV-15

Identificador: PV-16	
Título:	Eliminar una alerta.
Descripción:	Comprobar que se puede eliminar un *trigger* existente en el sistema para que desaparezca.
Requisitos relacionados:	RSF-61
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-15. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz.

Identificador: PV-16	
	4. Seleccionar “Templates” en la esquina superior derecha de la pantalla. 5. Seleccionar “OSL” en el menú desplegable “Group” 6. Seleccionar “Triggers” en la fila de “OSL_Linux”. 7. Buscar el *trigger* “SSH caído” en la lista y marcar la casilla junto a él. 8. Seleccionar “Delete selected” del menú desplegable de la parte inferior. 9. Seleccionar el botón “Go”.
Criterio de aceptación:	Comprobar que el *trigger* “SSH caído” ha desaparecido de la lista de *trigger*s de la plantilla “OSL_Linux”.

Tabla 164: Prueba de verificación PV-16

Identificador: PV-17	
Título:	Desactivar una alerta.
Descripción:	Comprobar que es posible dejar de emitir una alerta concreta para un host.
Requisitos relacionados:	RSF-62
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-15. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 4. Seleccionar “OSL” en el menú desplegable “Group” 5. Seleccionar “Triggers” en la fila de “osl-prox1”. 6. Buscar el *trigger* “SSH caído” en la lista y marcar la casilla junto a él. 7. Seleccionar “Disable selected” del menú desplegable de la parte inferior. 8. Seleccionar el botón “Go”.

Identificador: PV-17	
	9. Seleccionar la pestaña “Monitoring” y a continuación “Overview”. 10. Seleccionar “OSL” en el menú desplegable “Group”. 11. Seleccionar “Trigger” en “Type”.
Criterio de aceptación:	Comprobar que existe una fila “SSH caído” pero sin ningún cuadro de color (o color transparente) en la columna “osl-prox1”.

Tabla 165: Prueba de verificación PV-17

Identificador: PV-18	
Título:	Definición de una notificación.
Descripción:	Comprobar que se puede crear una nueva notificación que envíe mensajes cuando cambia el estado de un *trigger*.
Requisitos relacionados:	RSF-64, RSF-65, RSF-66, RSF-67, RSF-69, RSF-72
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-15. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña “Configuration” y a continuación “Actions” de la interfaz. 4. Seleccionar “Create Action”. 5. Establecer “Alerta SSH” en “Name”. 6. Seleccionar “Enabled” en el menú desplegable “Status”. 7. Seleccionar “New” en “Action condition”. 8. Seleccionar “Trigger” en el primer desplegable que aparece. 9. Seleccionar “=” en el segundo desplegable que aparece. 10. Seleccionar el botón “Select”. 11. Seleccionar “OSL” en el menú desplegable “Group”.

Identificador: PV-18	
	12. Seleccionar “OSL_Linux” en el menú desplegable “Host”. 13. Seleccionar “SSH caído”. 14. Seleccionar el botón “Add”. 15. Seleccionar “New” en “Action condition”. 16. Seleccionar “Trigger value” en el primer desplegable que aparece. 17. Seleccionar “=” en el segundo desplegable que aparece. 18. Seleccionar “OK” en el tercer desplegable que aparece. 19. Seleccionar el botón “Add”. 20. Seleccionar “And” en el menú desplegable “Type of calculation”. 21. Seleccionar el botón “New” en “Action operations”. 22. Seleccionar “Send message” en el menú desplegable “Operation type”. 23. Seleccionar el botón “Select” en “Send message to”. 24. Seleccionar el rol “Zabbix administrators”. 25. Seleccionar “-all-” en “Send only to” para enviar por e-mail, SMS y Jabber. 26. Seleccionar el botón “Add”. 27. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 28. Ejecutar el comando “/etc/init.d/ssh start”. 29. Ejecutar el comando “/etc/init.d/ssh stop”.
Criterio de aceptación:	Se recibirá un correo electrónico, un mensaje SMS y un mensaje instantáneo a la cuanta de Jabber, informando sobre la desactivación del servicio *ssh*.

Tabla 166: Prueba de verificación PV-18

Identificador: PV-19	
Título:	Desactivar notificación.
Descripción:	Comprobar que se puede dejar de emitir notificaciones

Identificador: PV-19	
	cuando se activa un *trigger* con una notificación definida.
Requisitos relacionados:	RSF-68
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-15. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña "Configuration" y a continuación "Actions" de la interfaz. 4. Seleccionar el enlace "Enabled" en la fila de "Alerta SSH". 5. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 6. Ejecutar el comando "/etc/init.d/ssh start". 7. Ejecutar el comando "/etc/init.d/ssh stop".
Criterio de aceptación:	No se recibe ningún tipo de notificación tras desactivar el servicio *ssh*.

Tabla 167: Prueba de verificación PV-19

Identificador: PV-20	
Título:	Eliminar notificación.
Descripción:	Comprobar que se puede eliminar una notificación definida en el sistema para que desaparezca y dejar de recibir mensajes cuando se activa su *trigger* asociado.
Requisitos relacionados:	RSF-70
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-15. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña "Configuration" y a continuación "Actions" de la interfaz. 4. Seleccionar la casilla junto a "Alerta SSH".

Identificador: PV-20	
	5. Seleccionar “Disable selected” del menú desplegable de la parte inferior. 6. Seleccionar el botón “Go”. 7. Comprobar que “Alerta SSH” no aparece en la pantalla “Actions”. 8. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 9. Ejecutar el comando “/etc/init.d/ssh start”. 10. Ejecutar el comando “/etc/init.d/ssh stop”.
Criterio de aceptación:	No se recibe ningún tipo de notificación tras desactivar el servicio *ssh*, y la acción “Alerta SSH” no aparece en la pantalla de acciones.

Tabla 168: Prueba de verificación PV-20

Identificador: PV-21	
Título:	Creación de plantilla.
Descripción:	Comprobar que se puede crear una plantilla como heredera de otra existente.
Requisitos relacionados:	RSF-73, RSNF-28
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Configuration” y a continuación “Host groups” de la interfaz. 3. Seleccionar el enlace “Templates” que aparece en la fila del grupo “OSL”. 4. Seleccionar el botón “Create template”. 5. Establecer “Plantilla Linux” en el campo “Name”. 6. Seleccionar el botón “Add” en “Link with template”. 7. Establecer el menú desplegable “Group” al valor “OSL”. 8. Marcar la casilla junto a “OSL_Linux”. 9. Seleccionar el botón “Select”.

Identificador: PV-21	
	10. Seleccionar el botón “Save”. 11. Seleccionar la pestaña “Configuration” y a continuación “Host groups” de la interfaz. 12. Seleccionar el enlace “Templates” que aparece en la fila del grupo “OSL”.
Criterio de aceptación:	Comprobar que existe una nueva plantilla llamada “Plantilla Linux” que tiene el mismo número de *item*s, *trigger*s y graphs que la plantilla “OSL_Linux”.

Tabla 169: Prueba de verificación PV-21

Identificador: PV-22	
Título:	Eliminación de plantilla.
Descripción:	Comprobar que es posible eliminar una plantilla definida en el sistema.
Requisitos relacionados:	RSF-74
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-21. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña “Configuration” y a continuación “Host groups” de la interfaz. 4. Seleccionar el enlace “Prueba Linux” que aparece en la fila del grupo “OSL”. 5. Seleccionar el botón “Delete”. 6. Seleccionar el botón “Aceptar” en el diálogo que aparece”.
Criterio de aceptación:	Comprobar que en la lista de plantillas ha desaparecido “Prueba Linux” y cualquier host que estuviese asociado a ella (continuarán esos host en el resto de plantillas existentes).

Tabla 170: Prueba de verificación PV-22

Identificador: PV-23	
Título:	Quitar plantilla.
Descripción:	Comprobar que es posible desvincular una plantilla de un host que la utiliza.
Requisitos relacionados:	RSF-76
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-21. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña "Configuration" y a continuación "Hosts" de la interfaz. 4. Seleccionar "Create host". 5. Introducir "host-prueba" en "name", su dirección IP en "IP address", "IP address" en "connect to". 6. Seleccionar "OSL" de la lista "Other groups" y pinchar sobre el botón "<<". 7. Seleccionar "Add" en "Linked templates". 8. Seleccionar "OSL" en "Group" y "Prueba Linux". 9. Seleccionar "Save". 10. Seleccionar la pestaña "Configuration" y a continuación "Host groups" de la interfaz. 11. Seleccionar el enlace "Templates" que aparece en la fila del grupo "OSL". 12. Comprobar que hay una fila en la tabla denominada "Prueba Linux" en cuya fila aparece el host "host-prueba". 13. Seleccionar el enlace "host-prueba". 14. Marcar la casilla junto a "Prueba Linux" en "Linked templates". 15. Pulsar el botón "Unlink" en "Linked templates". 16. Seleccionar el botón "Save".
Criterio de aceptación:	Comprobar que en la fila llamada "host-prueba" no hay ningún valor para la columna "Templates".

Tabla 171: Prueba de verificación PV-23

Identificador: PV-24	
Título:	Crear gráfica.
Descripción:	Comprobar que es posible definir una gráfica para un equipo o plantilla.
Requisitos relacionados:	RSF-77, RSF-79
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Configuration” y a continuación “Hosts” de la interfaz. 3. Seleccionar “OSL” en el menú desplegable “Group”. 4. Seleccionar el enlace “Graphs” en la fila “osl-prox1”. 5. Activar el botón “Create Graph”. 6. Introducir “Grafica prueba” en el campo “Name”. 7. Pulsar el botón “Add” junto a “Item”. 8. Seleccionar el botón “Select” del campo “Parameter”. 9. Seleccionar “OSL” en el menú desplegable “Group”. 10. Seleccionar “osl-prox1” en el menú desplegable “Host”. 11. Seleccionar el *item* “Available memory”. 12. Activar el botón “Add”. 13. Activar el botón “Save”. 14. Seleccionar la pestaña “Monitoring” y a continuación “Graphs”. 15. Seleccionar “OSL” en el menú desplegable “Group”. 16. Seleccionar “osl-prox1” en el menú desplegable “Host”. 17. Seleccionar “Grafica prueba” en el menú desplegable “Graph”.
Criterio de aceptación:	Comprobar que aparece en la pantalla una gráfica con los valores de memoria disponible en función del

Identificador: PV-24	
	tiempo.

Tabla 172: Prueba de verificación PV-24

Identificador: PV-25	
Título:	Eliminar gráfica.
Descripción:	Comprobar que es posible eliminar una gráfica definida para un equipo o plantilla.
Requisitos relacionados:	RSF-78
Procedimiento:	1. Realizar el procedimiento de la prueba de aceptación PV-24. 2. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 3. Seleccionar la pestaña "Configuration" y a continuación "Hosts" de la interfaz. 4. Seleccionar "OSL" en el menú desplegable "Group". 5. Seleccionar el enlace "Graphs" en la fila de "osl-prox1". 6. Activar la casilla junto a "Grafica prueba". 7. Seleccionar "Delete selected" en el menú desplegable de la parte inferior. 8. Activar el botón "Go". 9. Seleccionar la pestaña "Monitoring" y a continuación "Graphs". 10. Seleccionar "OSL" en el menú desplegable "Group". 11. Seleccionar "osl-prox1" en el menú desplegable "Host".
Criterio de aceptación:	Comprobar que en el menú desplegable "Graph" no aparece la opción "Grafica prueba".

Tabla 173: Prueba de verificación PV-25

Identificador: PV-26	
Título:	Disponibilidad de máquinas virtuales.
Descripción:	Comprobar la disponibilidad de una máquina virtual durante un día.
Requisitos relacionados:	RSNF-2
Procedimiento:	1. Abrir un cliente de escritorio remoto RDP. 2. Introducir la dirección IP (o nombre DNS) de una máquina virtual de teletrabajo. 3. Mantener la conexión durante 24 horas.
Criterio de aceptación:	Comprobar que la máquina virtual continua respondiendo tras haber pasado el ese periodo de tiempo.

Tabla 174: Prueba de verificación PV-26

Identificador: PV-27	
Título:	Control de acceso.
Descripción:	Comprobar que es imposible acceder al sistema desde fuera de la red de la Universidad y queda constancia de los intentos de acceso.
Requisitos relacionados:	RSNF-3, RSNF-12, RSNF-24
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh* desde un equipo externo a la red de la Universidad. 2. El cliente de *ssh* se queda bloqueado esperando la conexión o dice que es imposible realizar la conexión. Se termina el proceso. 3. Acceder mediante un cliente de escritorio remoto RDP a una máquina virtual introduciendo su dirección IP (o nombre DNS). 4. El cliente de escritorio remoto se queda bloqueado intentando acceder a la dirección o indica que no puede realizar la conexión. 5. Activar la conexión VPN en el equipo en el que se realizan las pruebas. 6. Acceder mediante un cliente de escritorio

Identificador: PV-27	
	remoto RDP a una máquina virtual introduciendo su dirección IP (o nombre DNS). 7. Cerrar la conexión de escritorio remoto. 8. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh*. 9. Ejecutar el comando “dmesg”.
Criterio de aceptación:	Comprobar que en el registro del sistema que sale, aparecen los 2 intentos de conexión fallidos y además funciona la conexión realizada con el cliente de escritorio remoto.

Tabla 175: Prueba de verificación PV-27

Identificador: PV-28	
Título:	Máquinas virtuales Windows monitorizadas.
Descripción:	Comprobar que se está monitorizando una máquina virtual.
Requisitos relacionados:	RSNF-4, RSNF-15
Procedimiento:	1. Acceder al servidor de monitorización mediante un navegador web introduciendo la dirección zabbixosl.uc3m.es e identificarse. 2. Seleccionar la pestaña “Monitoring” y a continuación “Overview” de la interfaz. 3. Seleccionar “Teletrabajo” en el menú desplegable “Group”. 4. Seleccionar “Triggers” en el menú desplegable “Type”.
Criterio de aceptación:	Comprobar que las casillas de la tabla de la fila “Server {HOSTNAME} is unreachable” están en verde.

Tabla 176: Prueba de verificación PV-28

Identificador: PV-29	
Título:	Puntos de restauración de un backup.
Descripción:	Comprobar el número de backups que se pueden

Identificador: PV-29	
	restaurar.
Requisitos relacionados:	RSNF-19
Procedimiento:	1. Acceder al servidor de Teletrabajo prox1 mediante una sesión remota con *ssh* desde un equipo externo a la red de la Universidad. 2. Ejecutar el comando "/usr/local/bin/vmtools/vmbackup –r 111" para ver los backups disponibles de la máquina 111.
Criterio de aceptación:	Comprobar que se muestran al menos dos backups en la pantalla con fechas inferiores a siete días.

Tabla 177: Prueba de verificación PV-29

Identificador: PV-30	
Título:	Recuperación de máquina virtual.
Descripción:	Comprobar que se puede acceder a una máquina virtual caída en menos de un día laborable.
Requisitos relacionados:	RSNF-21
Procedimiento:	1. Acceder al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 2. Ejecutar el comando "/usr/local/bin/vmtools/vmmanagement stop 101 101". 3. Acceder mediante un cliente de escritorio remoto RDP a la máquina virtual 101 introduciendo su dirección IP (o nombre DNS). 4. El cliente de escritorio remoto se queda bloqueado intentando acceder a la dirección o indica que no puede realizar la conexión. 5. Esperar hasta que sea la misma hora del próximo día laborable. 6. Acceder mediante un cliente de escritorio remoto RDP a la máquina virtual 101 introduciendo su dirección IP (o nombre DNS).

Identificador: PV-30	
Criterio de aceptación:	Comprobar que la máquina virtual está funcionando.

Tabla 178: Prueba de verificación PV-30

Identificador: PV-32	
Título:	Privacidad de sesión.
Descripción:	Comprobar que cuando un teletrabajador está haciendo uso de su máquina virtual, no se puede visualizar su actividad.
Requisitos relacionados:	RSNF-26
Procedimiento:	1. Acceder mediante un cliente de escritorio remoto RDP a la máquina virtual 101 introduciendo su dirección IP (o nombre DNS). 2. Se accede al servidor de Teletrabajo mediante un navegador web a la dirección prox1.uc3m.es. 3. Se selecciona “Máquinas virtuales” en el menú de la izquierda. 4. Se selecciona la máquina virtual 101. 5. Se selecciona “Open VNC console” para abrir la consola de la máquina.
Criterio de aceptación:	Comprobar como en la consola VNC accedida a través de la interfaz de Teletrabajo, solo se puede visualizar que la sesión ha sido bloqueada.

Tabla 179: Prueba de verificación PV-32

Identificador: PV-33	
Título:	Autenticación de servidores.
Descripción:	Comprobar que los servidores de Teletrabajo no pueden ser reemplazados al autenticarse frente al usuario.
Requisitos relacionados:	RSNF-27
Procedimiento:	1. Acceder al servidor de Teletrabajo mediante un

Identificador: PV-33	
	navegador web a la dirección prox1.uc3m.es. 2. Usar la opción de visualización del certificado del navegador.
Criterio de aceptación:	Comprobar que la dirección que aparece en el navegador, es la misma que la que aparece en el certificado en el campo "Dueño".

Tabla 180: Prueba de verificación PV-33

Identificador: PV-34	
Título:	Servidor de monitorización independiente.
Descripción:	Comprobar que el servidor de monitorización está localizado en una máquina física distinta de los servidores de Teletrabajo.
Requisitos relacionados:	RSNF-32
Procedimiento:	1. Acceder al servidor de monitorización zabbixosl mediante una sesión remota con *ssh* desde un equipo externo a la red de la Universidad. 2. Reiniciar o apagar los servidores prox1 y prox2.
Criterio de aceptación:	Comprobar que la sesión remota con zabbixosl no se pierde.

Tabla 181: Prueba de verificación PV-34

5.2. Pruebas de rendimiento

Durante la fase de pruebas del servicio de Teletrabajo, se realizaron una serie de pruebas de carga en todas las máquinas virtuales, simultáneamente, mediante la herramienta *HeavyLoad*, tomando mediciones en los clústeres del servicio de Teletrabajo con *atop*. Tras analizar los datos obtenidos de uso de CPU y escritura en disco, se descubrió que, pese a no tener una carga elevada, las máquinas virtuales tenían un rendimiento muy bajo. Estos problemas se consiguieron subsanar cambiando los dispositivos de entrada/salida, virtualizados en las máquinas virtuales (adaptador de red y controlador de disco duro). Todo el procesamiento y mediciones realizados se pueden consultar en el proyecto previo (Gil Bázquez, 2011).

Una vez en la fase de producción del servicio de Teletrabajo, gracias a las mediciones realizadas por el servicio de monitorización, se detectó que el uso del espacio de almacenamiento de algunas máquinas virtuales estaba siendo más alto que el previsto inicialmente. En concreto, el espacio libre de algunas máquinas llegó a bajar por debajo de un gigabyte en las primeras semanas de la fase de producción.

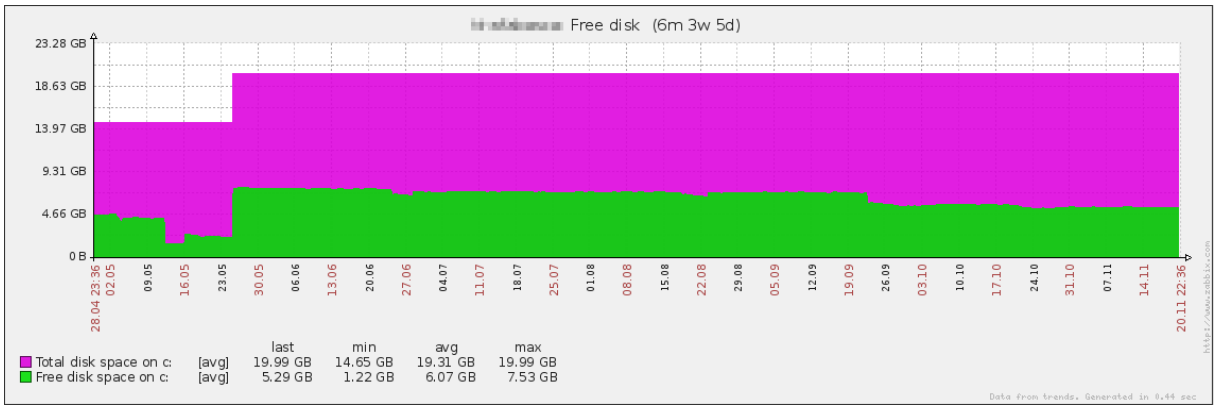

Ilustración 112 Uso del disco duro de máquina virtual

Se llevó a cabo un estudio de la capacidad de almacenamiento disponible en el disco en red iSCSI y del espacio local en los servidores de Teletrabajo, para el almacenamiento de backup. Como resultado se decidió realizar una parada programa de mantenimiento del servicio, para incrementar el tamaño del disco duro de las máquinas virtuales. Durante la parada de mantenimiento se amplió en 5 GB la capacidad de almacenamiento de cada máquina virtual, pasando de 15 GB de disco duro a 20 GB por máquina.

Tras el funcionamiento ininterrumpido del servidor zabbixosl durante varios días, se detectó, durante un uso normal de su interfaz web, que los tiempos de carga de las páginas, en especial aquellas que deben cargar gráficos, son elevados en comparación con los tiempos del servidor recién arrancado. No sólo eso, algunas máquinas no presentaban valores actualizados en tiempo real, sino que tardaban unos minutos en actualizar una serie de valores. Para poder detectar a qué se debía este comportamiento, se recurre a la herramienta *htop*, la cual proporciona, de una manera gráfica, información sobre el uso de CPU, memoria y los procesos que ejecuta una máquina.

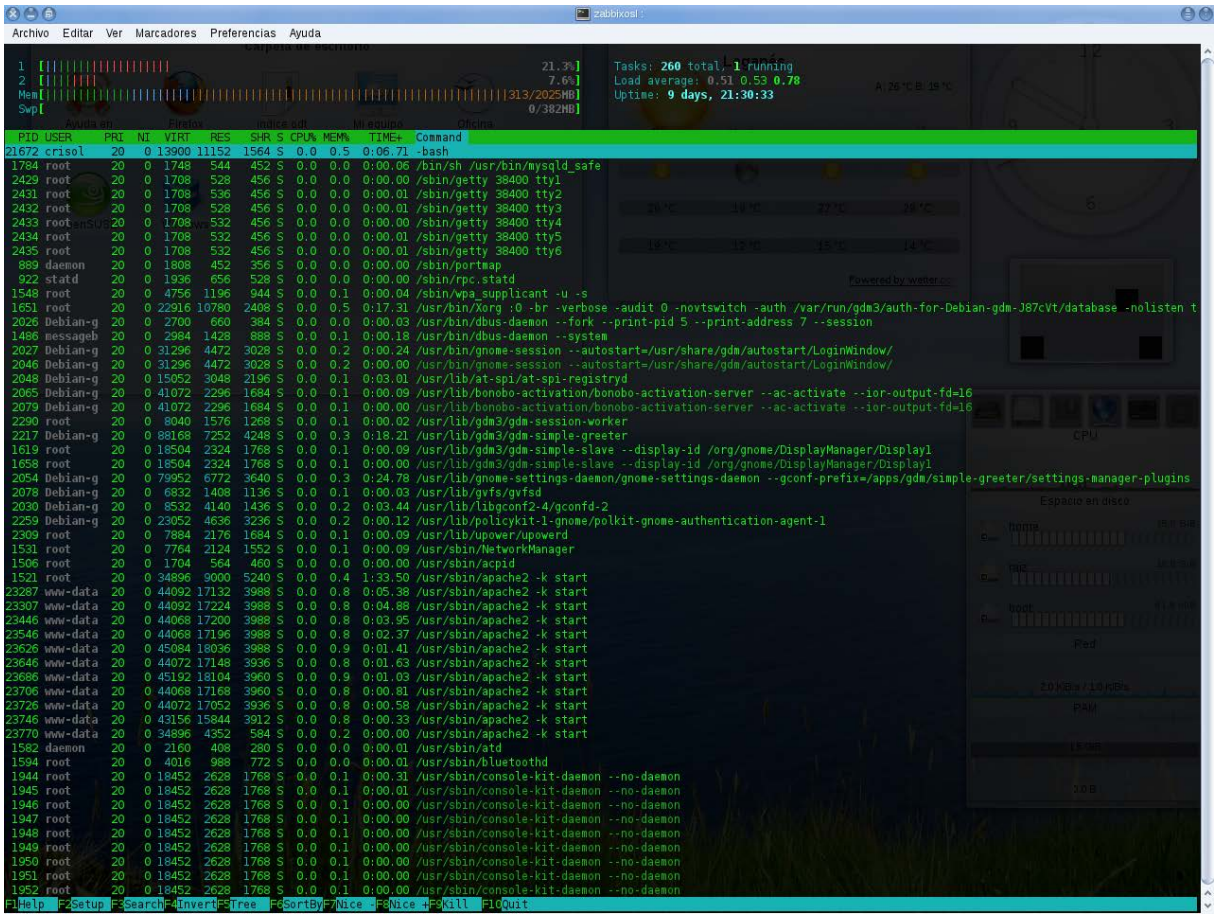

Ilustración 113 Medición con htop del rendimiento del servidor de monitorización

En la *Ilustración 113 Medición con htop del rendimiento del servidor de monitorización* se muestran los resultados que nos proporciona la ejecución de *htop*, al mismo tiempo que en un navegador se cargan múltiples gráficas, el trabajo más pesado que se puede pedir al servidor de monitorización. Sorprendentemente, según *htop*, el consumo tanto en ambas CPUs como de memoria es muy bajo, de lo que se dedujo que se trataba de un problema de acceso a disco o base de datos. Puesto que la base de datos MySQL no había sido optimizada ni configurada para los requisitos de rendimiento que puede presentar un sistema con uso tan intensivo de base de datos como Zabbix. Era de esperar que los problemas de rendimiento fueran originados por la base de datos. De este modo se recurrió a la herramienta MySQL-tuner, que tiene como requisito que la base de datos lleve al menos 24 horas en ejecución. Este requisito se cumplía en el momento de realizar la optimización, pero impone una limitación a la hora de afinar un ajuste óptimo de manera iterativa, ya que hay que esperar un día entero entre cada ajuste para comprobar si funciona correctamente. Tras lanzar el script *mysqltuner* y proporcionarle cuando lo solicita el usuario "root" de la base de datos y su contraseña, proporcionó la salida reflejada en la siguiente figura.

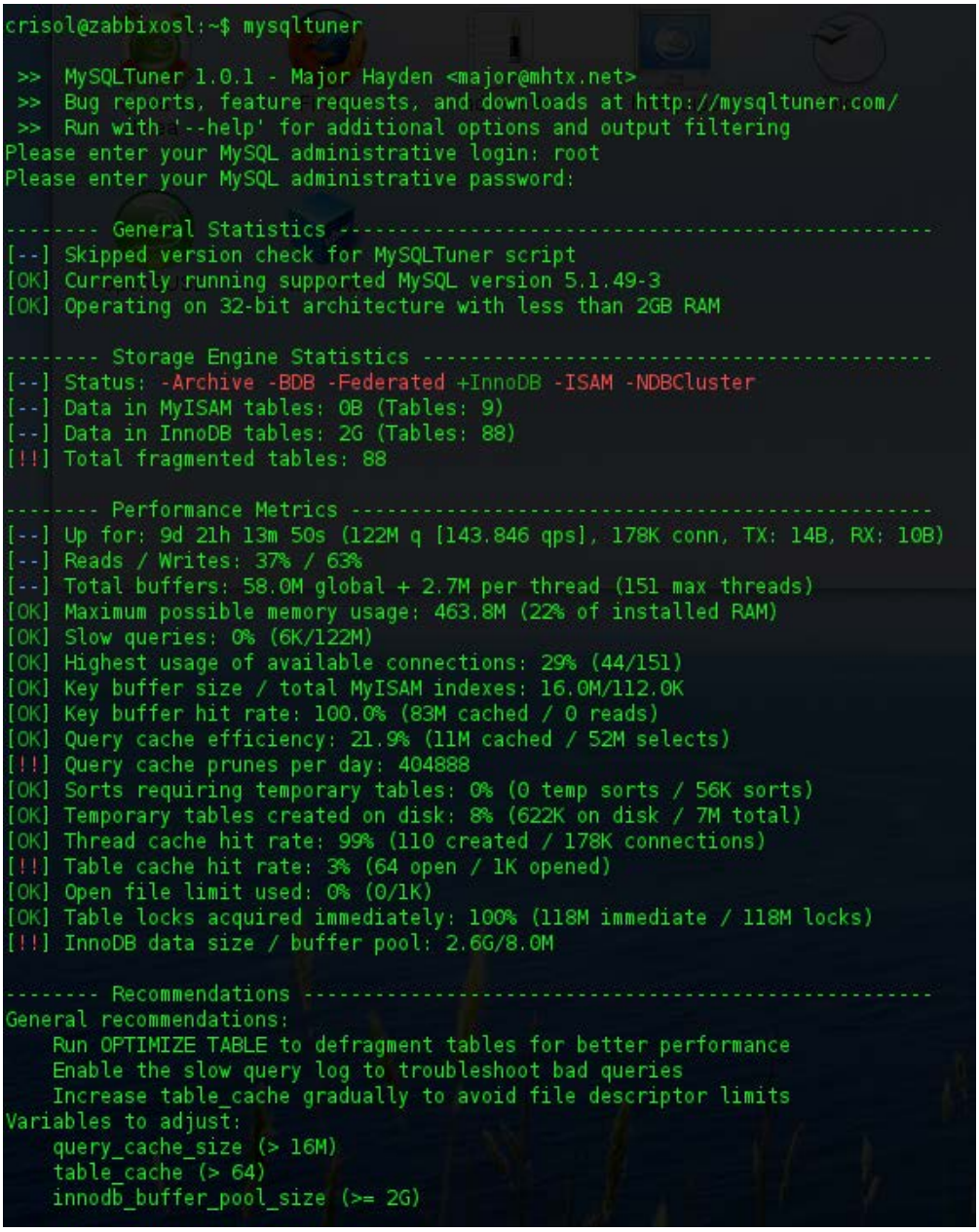

Ilustración 114 Primera ejecución de MySQL-tuner

MySQL-tuner proponía (como se aprecia en *Ilustración 114 Primera ejecución de MySQL-tuner*) aumentar los valores de las propiedades "query_cache_size", "table_cache" e "innodb_buffer_pool_size", proporcionando unos valores mínimos, aunque no unos máximos o recomendados. Como primera aproximación se decide utilizar el doble del tamaño mínimo para "query_cache_size" y "table_cahe", para innodb_buffer_pool_size nos sugiere un mínimo de 2 gigabytes, puesto que es el tamaño de la memoria RAM de la máquina, no se le debería dar un valor mayor, incluso utilizar 2GB no sería tampoco recomendable, no obstante, dado que es el valor mínimo que sugiere que se le dé, se acepta ese valor. Para establecer los nuevos valores, hay que editar el fichero "/etc/mysql/my.cnf" mediante un editor como puede ser Vim, buscar las

propiedades descritas y modificarlas o crearlas si no existen, en la *Tabla 182 Resumen de ajustes de MySQL* se ofrece un resumen de los cambios modificados.

Parámetro	Valor original	Valor nuevo
query_cache_size	16M	32M
table_cache	sin valor	128
innodb_buffer_pool_size	sin valor	2G

Tabla 182 Resumen de ajustes de MySQL

Además se sugerían otras recomendaciones que, un principio no se aplican, ya que una requieren modificar el script de arranque de MySQL (lo que puede ser peligroso), pero que serían tenidas en cuenta como segundo paso a realizar si los primeros ajustes no consiguen mejorar el rendimiento de la interfaz web de Zabbix y de actualizar los valores de los *item*s. La desfragmentación de las tablas, aunque tediosa, es una tarea bastante sencilla. Se realizó antes de modificar ninguna propiedad, puesto que es el cambio menos agresivo.

Una vez guardados los cambios y reiniciado el servidor de MySQL, se volvió a probar la navegación por la interfaz web de Zabbix, comprobando que, efectivamente, se apreciaba una mejora significativa en el tiempo en la carga de cada una de las páginas, en especial, aquellas que presentaban varios gráficos, que podían tardar hasta minutos en cargar, ahora lo hacían de manera instantánea.

Para asegurar que la base de datos estaba correctamente configurada, se volvió a lanzar MySQL-tuner, que advirtió que la cantidad de memoria RAM utilizada actualmente por MySQL era alta, en comparación con la memoria de la máquina (2GB). Se recurrió a *htop* para comprobar si es cierta dicha advertencia, obteniendo los resultados de la *Ilustración 115 Medición con htop del rendimiento del servidor de monitorización tras el ajuste*.

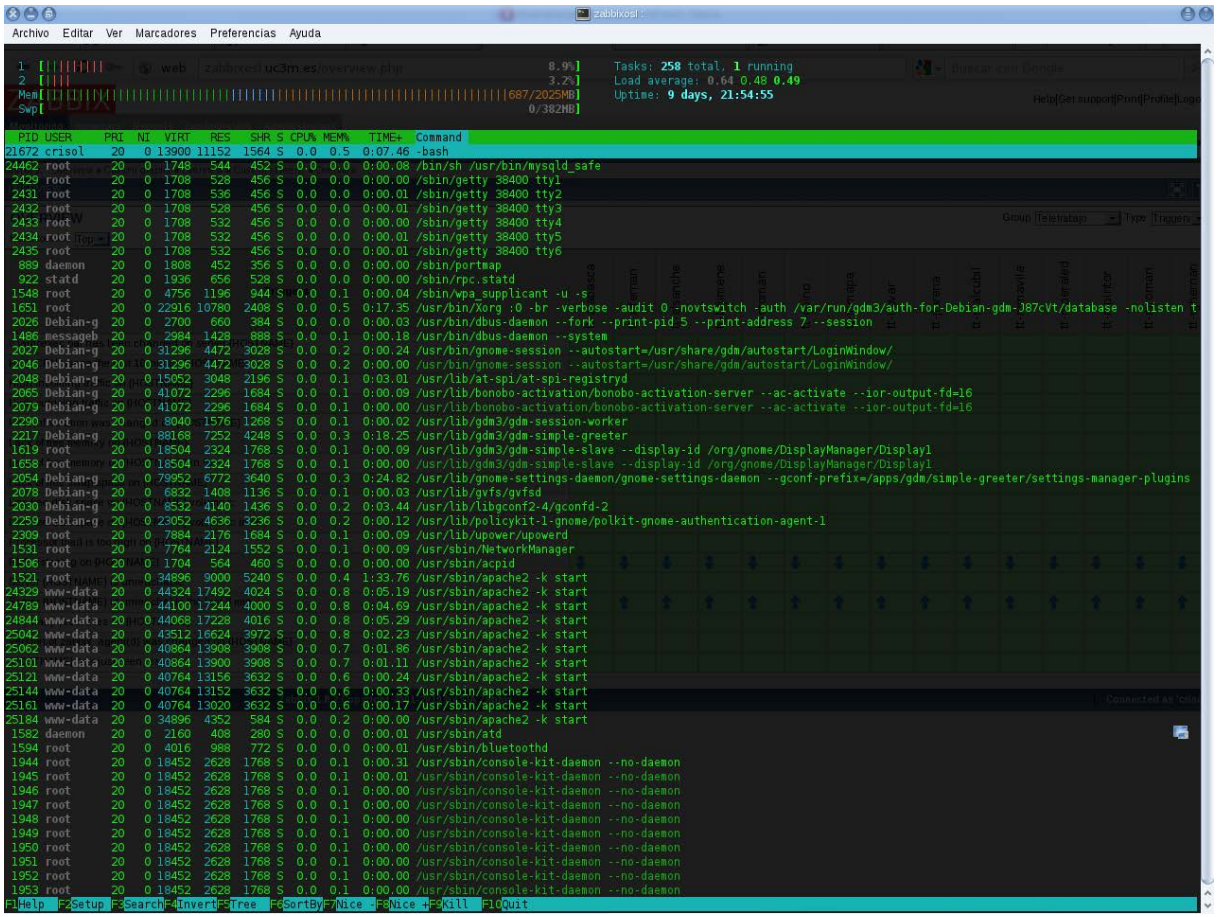

Ilustración 115 Medición con htop del rendimiento del servidor de monitorización tras el ajuste

Htop mostró, como era de esperar, un uso de memoria RAM más elevado que anteriormente, concretamente 687MB. Este valor teniendo en cuenta que la máquina cuenta con 2GB de memoria RAM, y no está llegando a utilizar memoria de intercambio, no parece motivo de preocupación, como sugería el script MySQL-tuner. No obstante, es cierto que se ha asignado un buffer de 2GB, que en casos extremos podría llegar a saturar la memoria (aunque tengamos memoria de intercambio para evitarlo). En un servidor de monitorización, como es el caso, no se puede permitir correr el riesgo de dejar inutilizada la máquina, lo que lleva a plantear 2 soluciones: ampliar la memoria RAM de la máquina o reducir el tamaño de las caches de MySQL. Dado que aumentar la cache de la base de datos aportó una mejora tan notable, y puesto que una de las ventajas de las máquinas virtuales es que es muy sencillo cambiar el hardware de las mismas, se optó por la primera opción, asignando a la máquina 3GB de memoria RAM.

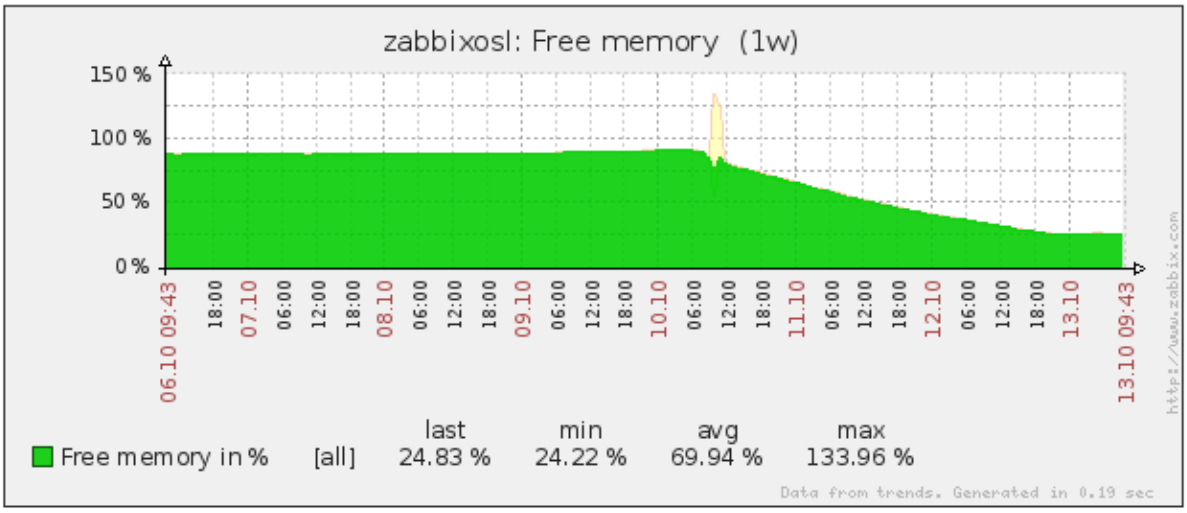

Ilustración 116 Uso de memoria del servidor de monitorización tras la optimización

A los dos días, tras ver que el gráfico de ocupación de memoria se había estabilizado, se considera que ha pasado tiempo suficiente para pasar una segunda prueba y, a pesar de que el rendimiento de la base de datos no se había visto reducido como ocurría en un principio, se retomó el procedimiento realizado con MySQL-tuner, obteniendo los datos de la *Ilustración 117 Segunda ejecución de MySQL-tuner.*

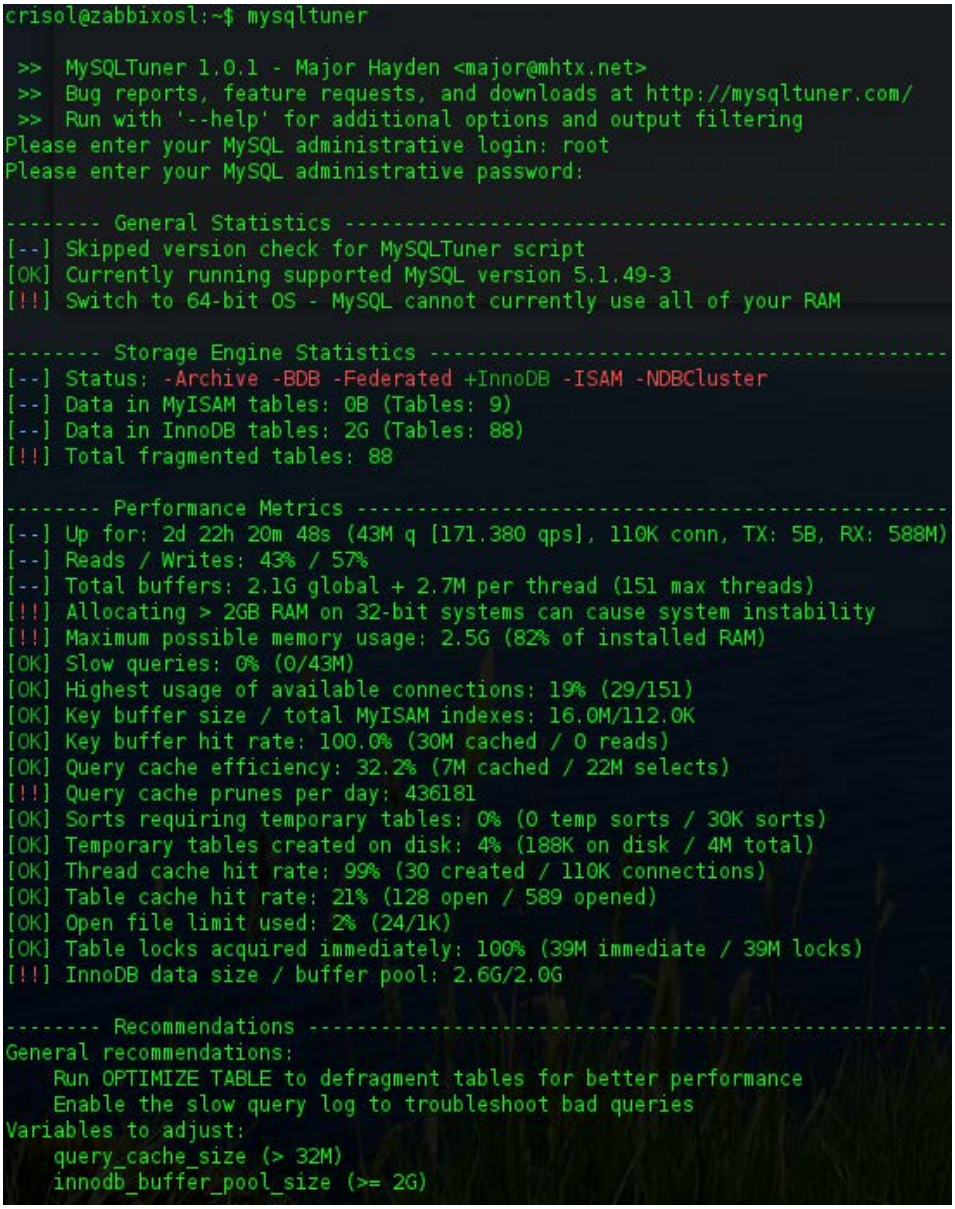

Ilustración 117 Segunda ejecución de MySQL-tuner

La ejecución del script volvió a sugerir incrementar el tamaño de query_cache_size a más de 32M. También indicó que se incrementara innodb_buffer_pool_size a al menos 2GB, pero ya tenía ese valor y, vistos los datos de ocupación de memoria de *Ilustración 116 Uso de memoria del servidor de monitorización tras la optimización*, no era recomendable incrementarlo aún más. Finalmente, se realizaron los cambios expuestos en *Tabla 183 Resumen del segundo ajuste de MySQL* en el fichero "my.cnf" y se reinició el servicio de MySQL para aplicar los nuevos cambios.

Parámetro	Valor original	Valor nuevo
query_cache_size	32	64

Tabla 183 Resumen del segundo ajuste de MySQL

Capítulo VI

Conclusiones

Una vez finalizado el desarrollo del proyecto es preciso verificar que se han cumplido con éxito los objetivos que se establecieron al inicio del mismo.

Durante el proyecto se ha diseñado una política de control de acceso, que ha sido implementada utilizando el módulo de filtrado de paquetes del núcleo de Linux, iptables, y configurado mediante la herramienta Shorewall. La configuración de iptables evita a cualquier equipo externo a la red de la universidad poder conectarse a los servidores de Teletrabajo. De modo que los servidores quedan protegidos de posibles atacantes que se pudieran encontrar en Internet, permitiendo el acceso a los usuarios y administradores del servicio, pudiendo concluir que se ha cumplido el objetivo de control de acceso. No obstante, la decisión de incrementar la seguridad, permitiendo únicamente conexiones desde la red de la UC3M, ha provocado que los teletrabajadores necesiten configurar una red VPN en sus equipos, aspecto que afecta a la transparencia de los servicios de seguridad de cara al usuario.

Para realizar las copias de seguridad se han creado una serie de *scripts* de *backup*, que se encargan de almacenar una instantánea de los discos duros de las máquinas virtuales en un disco de los servidores de teletrabajo dedicado a esta función. La ejecución de los *scripts* se ha automatizado utilizando la herramienta *cron*, que nos proporciona el sistema GNU/Linux instalado, así se consigue que los backups se realicen periódicamente dos veces a la semana, eliminando la necesidad de que se tenga que encargar un administrador de realizarlo. Además una vez a la semana, se realiza un segundo tipo de backup, mediante el servicio corporativo que ofrece la Universidad. Con este tipo de backup, conseguimos que los datos estén replicados en tres servidores diferentes: el disco de red donde está localizado el disco duro de las máquinas virtuales, el disco duro de los servidores de Teletrabajo, y la cinta donde se almacena el backup del servicio corporativo, quedando cubierto así la parte de replicación de datos del objetivo de disponibilidad, y totalmente satisfecho el objetivo de copias de seguridad.

En los scripts de *backup* desarrollados y programados mediante *cron*, se ha añadido la funcionalidad de realizar un backup cruzado, consistente en la copia de las máquinas virtuales pertenecientes a cada nodo del clúster de teletrabajo al otro nodo. Permitiendo, gracias a esta funcionalidad, que si falla cualquiera de los nodos del clúster, las máquinas virtuales sigan activas al estar respaldadas por el otro nodo del clúster. Si además añadimos el plan de contingencia detallado en *4.7 Mantenimiento del proyecto*, donde se explican los procedimientos a seguir para subsanar las incidencias que puedan surgir, incluyendo las instrucciones para activar las máquinas virtuales replicadas, quedan cubiertas todas las necesidades para poder afirmar que se ha alcanzado el objetivo de disponibilidad.

Finalmente, se ha creado un servidor dedicado a la monitorización de otros equipos, en el que se ha implantado la herramienta Zabbix para llevar a cabo ese objetivo. Para realizar esta monitorización, se ha desplegado un pequeño programa llamado agente de Zabbix en cada una de las máquinas que se quieren monitorizar, encargado de la realización de mediciones en los equipos y transmisión de los datos al servidor de monitorización. Para llevar a cabo esta monitorización se han desarrollado una serie de plantillas adaptadas a la monitorización de distintos tipos de equipos, dotando al servicio de una mayor coherencia y simplicidad en el momento de añadir nuevos equipos a monitorizar.

El servidor de monitorización implantado, se encarga de mantener un histórico con todos los valores medidos recopilados y mostrarlos en forma de tablas o gráficas. Estas gráficas muestran el rendimiento de los equipos y, a partir de las curvas descritas, permiten hacer predicciones sobre las necesidades de recursos de las máquinas virtuales de teletrabajo, y poder actuar en consecuencia ajustando las máquinas a las necesidades del servicio.

Gracias a la supervisión de distintos parámetros de los equipos por el servicio de monitorización, se han podido establecer unos umbrales críticos que permiten la detección de problemas, antes incluso de que lleguen a producirse. Estos umbrales se han asociado con diversas alertas que notifican del peligro por diversos modos: desde una pantalla de incidencias, a mensajes por correo electrónico. Esta supervisión, las 24 horas del día, y la inmediata notificación ante cualquier indicio de problema, ha permitido acortar los tiempos de reacción ante caídas de los equipos o incluso llegar a evitarlas.

Como se ha mencionado durante el documento. El sistema de monitorización a sido utilizado para monitorizar otros servidores de la Oficina de Software Libre, de hecho, en el momento de finalizar este documento se están monitorizando doce equipos ajenos al servicio de Teletrabajo.

Haber logrado cumplir todos los objetivos expuestos anteriormente hace que, mediante la suma de todos ellos, se pueda llegar a la conclusión de haber conseguido cumplir el objetivo principal que justificó el desarrollo de este proyecto. Se ha conseguido dotar de **seguridad** al servicio de Teletrabajo de la Universidad Carlos III de Madrid.

Capítulo VII

Líneas futuras

El objetivo de este apartado es la identificación y propuesta de mejoras futuras al sistema desarrollado en el presente proyecto.

Puesto que la funcionalidad que ofrecía Proxmox a través de su interfaz no ha llegado a cubrir todas nuestras necesidades de administración, como es el caso del backup, que ha requerido el uso de scripts poco integrados con Proxmox, lo que dificulta las tareas de administración. Una solución ideal sería el desarrollo de una nueva interfaz web o, en su defecto, ampliar la existente para poder lanzar la ejecución de los scripts cómodamente, con ayuda del ratón y sin necesidad de tener que recurrir a introducir comandos de texto, difíciles de recordar en una consola.

Sin duda, la funcionalidad más limitada ha sido la de backup. De contar con mejor hardware de almacenamiento, tanto en capacidad como en velocidad de escritura y lectura, se podría incrementar el número de puntos de restauración por ciclo de las máquinas virtuales, e incluso mejorar su eficiencia utilizando alguno de los algoritmos diseñados para el backup.

También podría ser aumentada la eficiencia del backup, realizándolo de los ficheros en lugar de tratar con discos binarios, como se hace actualmente. Realizar un backup de los ficheros del disco, además de mejorar la eficiencia del backup, se podría tener un control de versiones de los ficheros, de forma incremental respecto a un sistema de ficheros base, que ofrecería prácticamente ilimitados puntos de restauración de la máquina. La implantación de este tipo de backups requeriría de la implantación de un nuevo servidor de backup, y agentes de backup instalados en las máquinas virtuales que, a través de su conexión de red, contactasen con el servidor de backup para transmitir los datos.

El desarrollo de semejante infraestructura de backup supondría una serie de retos adicionales, desde la restauración de un backup (ya que no bastaría con remplazar el disco de

la máquina por el del backup), hasta retos de seguridad, al ser necesario asegurar la confidencialidad de los datos transmitidos y almacenados.

En cuanto al sistema de monitorización, se ha mencionado durante el documento que también va a ser utilizado para monitorizar otros servidores de la Oficina de Software Libre. No solo se está limitando a la monitorización de equipos sino que, gracias las capacidades de monitorizar sitios web de Zabbix, se está supervisando la disponibilidad de la página principal del portal web de la Oficina de Software Libre. Aunque se esta realizando una monitorización, no se hace de forma completa, quedando supervisar cada una de las páginas del portal de la OSL y emitir notificaciones si por ejemplo se rompe algún enlace, o el tiempo de carga de la página web comienza a ser muy elevado.

También, gracias al sistema de monitorización, actualmente se han automatizado algunas tareas de mantenimiento, como por ejemplo volver a lanzar el servicio de ssh si detecta que no está activo. Sin embargo, se podría aumentar la disponibilidad de las máquinas virtuales si se automatiza su recuperación ante caídas, mediante comandos remotos en los servidores de Teletrabajo, dejando de depender de un administrador que reciba la notificación y levante las máquinas manualmente. Esta automatización no ha sido posible realizarla en este proyecto, al necesitar un mecanismo lo suficientemente fiable para detectar la caída de las máquinas, ya que un falso positivo podría provocar la pérdida de sesión del usuario al forzar un re-arranque automático. Actualmente el mecanismo con el que se detecta el funcionamiento de una máquina es que esta responda a una petición "ping" del servidor Zabbix, sería posible implementar otro método pero por la dificultad que suponía se ha excluido del proyecto actual.

Las pantallas de la interfaz de Zabbix son de un tamaño grande puesto que presentan mucha información en una única página. Esta situación hace que sea especialmente difícil utilizar el servicio de monitorización desde un dispositivo móvil o con pantalla de poca resolución. Debido a lo interesante que puede resultar consultar el estado de los equipos monitorizados desde cualquier lugar, por ejemplo para supervisar una tarea de mantenimiento, de modo que una importante mejora sería el desarrollo de una interfaz para dispositivos móviles.

Por último, también referente al sistema de monitorización, sería interesante el desarrollo de *item*s que realicen sus mediciones mediante el protocolo SNMP, que se ha dejado fuera del proyecto debido a su complejidad. Con estos nuevos *items*, se podrían monitorizar equipos en la red que no cuenten con un agente de monitorización instalado, permitiéndonos, de este modo, monitorizar nodos de la red de telecomunicaciones (como enrutadores, puentes, conmutadores). Conocer el estado de estos puntos de la red, aportaría información importante para detectar posibles puntos de fallo ante cualquier incidencia.

Capítulo VIII

Glosario

API: (Application Programming Interface) Interfaz de programación de aplicaciones en español, es un conjunto de funciones y procedimientos ofrecidos por la biblioteca de un programa para que sean utilizados por un segundo programa. Proporcionan un mayor nivel de abstracción y facilitan al programador el desarrollo del software.

Array: Un array es una estructura de datos que permite almacenar un conjunto de elementos de un mismo tipo. Cada elemento almacenado consta de dos partes: el dato que se desea almacenar y un índice que lo identifica inequívocamente dentro de la estructura.

CMS: (Content Management System) Sistema de Gestión de Contenidos en español, software que crea una plataforma para la creación y administración de contenido, normalmente páginas webs, en un entorno colaborativo.

Cron: demonio de los sistemas Unix que se encarga de ejecutar procesos en intervalos regulares de tiempo. Permite definir mediante ficheros de configuración cuáles son los procesos que se desean lanzar en una máquina, indicando el momento exacto (año, mes, día, día de la semana, minuto etc.) y la periodicidad con la que se deben ejecutar.

Demonio: Daemon en inglés (Disk And Execution MONitor), es el nombre que reciben en informática los procesos que se ejecutan en segundo plano en el sistema, sin intervención del usuario y de forma continuada.

DNS: (Domain Name Service) Servicio de Nombres de Dominio en español. Protocolo jerárquico distribuido para dar nombres a sistemas informáticos. Su objetivo es proporcionar una traducción textual y entendible por humanos a las direcciones IP numéricas que usan los ordenadores.

Emacs: (Editor MACroS) Editor de texto parte del proyecto GNU, es uno de los editores más potentes actualmente por la gran cantidad de utilidades para manejar textos. Uno de sus características más potentes es la capacidad de ampliarlo añadiendo nueva funcionalidad.

Framework: Colección de bibliotecas software que proveen una API.

Freeware: Tipo de software que se distribuye gratuitamente o a cambio de un pago opcional, pero con al menos algún derecho restringido: copia, modificación, redistribución o uso. Normalmente se refiere a software gratuito de código cerrado, aunque puede referirse también a software de código abierto.

Gzip: (GNU ZIP) herramienta del proyecto GNU para comprimir archivos mediante el algoritmo de deflación (algoritmo de compresión sin pérdidas). Surgió en 1992 como una alternativa libre al programa *compress* de Unix.

HTML: (HyperText Markup Language) Lenguaje de marcado utilizado para la elaboración de páginas web, definiendo la presentación de documentos.

Inetd: Proceso de sistemas Unix encargado de agrupar varios demonios, haciendo que su ejecución sea más eficiente que la de los demonios por separado. El proceso escucha las conexiones entrantes del equipo e invoca el demonio adecuado en cada caso.

IP: (Internet Protocol) Es el protocolo principal utilizado para el intercambio de paquetes de datos a través de una red. IP permite la entrega de paquetes de datos mediante únicamente la dirección del destinatario.

Imagen ISO: Fichero que contiene una imagen exacta de un sistema de ficheros, almacenado según la especificación de la norma ISO 9660, del que obtiene el nombre.

JavaScript: Lenguaje de programación de alto nivel, interpretado, débilmente tipado y orientado a objetos. Es utilizado en páginas web para dotarlas de contenido dinámico, el código se interpreta en la máquina del cliente, por lo que debe contar con un intérprete de éste lenguaje.

KVM: (Kernel-based Virtual Machine) Plataforma de virtualización completa en hardware con arquitecturas x86. Consta de dos partes: un módulo del núcleo de Linux (incluido a partir de la versión 2.6.20) y aplicaciones de usuario, como el visor de máquinas virtuales Qemu que utiliza. Es una solución de virtualización completamente libre.

Lenguaje de marcado: Sistema para de realizar anotaciones en un texto, de tal manera que sean sintácticamente diferenciables del resto del texto en sí. Existen tres tipos de marcado: de presentación, en el que se indica cómo se debe visualizar el texto mediante las marcas; procedural, que provee instrucciones para el procesado del texto por un programa; y descriptivo, que añade semántica a los términos marcados del texto.

Lenguaje de programación de alto nivel: Lenguaje de programación que abstrae al programador de los detalles de la máquina, haciendo que sea más sencillo de entender para un ser humano que el código que manejan las máquinas.

Lenguaje débilmente tipado: Lenguaje que no comprueba o controla el tipo de datos que se almacena en las variables, puesto que no tiene por qué ser definido.

Lenguaje interpretado: Lenguaje de programación que es ejecutado por un intérprete. Debido a que los programas no se generan como código de bajo nivel, pueden ser ejecutados a partir del código fuente programado.

Log: O ficheros de log, son ficheros creados y mantenidos por una máquina, que contienen un registro de la actividad realizado por ella. Son utilizados, tanto en depuración como en seguridad, para conocer los detalles de cómo y cuándo ha surgido un determinado evento en el sistema.

NTFS: (New Technology File System) Sistema de ficheros que utiliza el sistema operativo Windows NT basado en los sistemas de ficheros HFS de Apple y HPFS de IBM. Es utilizado en sistemas con gran tamaño de disco duro (hasta 16 TB) que exigen un gran rendimiento, por ejemplo, servidores.

Programación orientada a objetos: Paradigma de programación en el que los programas se representan mediante entidades denominadas "objetos", que se relacionan entre ellas mediante los tipos de relación: herencia, polimorfismo, abstracción y encapsulamiento.

Prompt: Carácter mostrado en la línea de comandos para indicar al usuario que espera por una orden. Generalmente en Bourne Shell y sus derivado, se utiliza el carácter '$' o '#' en caso del súper-usuario.

Perl: Lenguaje de programación de alto nivel orientado a objetos, originalmente ideado para procesar textos. Combina elementos de sintaxis de lenguajes como C, con la de la shell de Unix. Es un proyecto de software libre iniciado en 1987.

Proxy: Programa o máquina que hace de intermediario en una comunicación entre otras dos entidades. El cliente se conecta al proxy para solicitar un servicio, el proxy, si no puede subsanarlo por sí mismo, se conecta con un servidor que sí pueda hacerlo, obteniendo el resultado del mismo que retransmite al cliente que inició la comunicación.

RDP: (Remote Desktop Protocol) Protocolo desarrollado por Microsoft utilizado para ofrecer una interfaz gráfica a una máquina distinta, permitiendo el control remoto de la máquina. El protocolo es una extensión del protocolo estandarizado en ITU-T T.128 para el protocolo para compartir aplicaciones.

Rootkit: Herramientas diseñadas para esconderse a ellas mismas, y otros programas, dedicadas a extraer información o permitir la manipulación ilegitima de un equipo sin ser detectado.

Scp: (secure Copy) Es el nombre del protocolo y el programa que lo implementa, encargados de la transmisión segura de ficheros entre un equipo cliente y otro servidor, utilizando para ello una conexión segura ssh.

Script: Programas, normalmente sencillos, almacenados e interpretados en forma de texto plano.

Shell: Software que ofrece una interfaz para comunicarse con el núcleo del sistema operativo. Existen principalmente dos tipos de shell: interfaz de línea de comandos (CLI en inglés) e interfaz gráfica de usuario (GUI en inglés).

SNMP: (Simple Network Management Protocol) Protocolo estandarizado por la IETF (Internet Engineering Task Force), parte de la Internet Protocol Suite, de nivel de aplicación. Sirve para gestionar dispositivos en una red TCP/IP y monitorizarlos.

Ssh: (Sechure SHell) Protocolo y programa que lo implementa, permite el acceso remoto a una máquina ofreciendo un intérprete de comandos y la transmisión de ventanas. La principal ventaja de ssh es que la conexión realizada es segura, siendo confidenciales los datos transmitidos gracias a los protocolos de cifrado que implementa.

Tar: **(**Tape ARchiver) Tipo de fichero que se utiliza para archivar varios ficheros en uno único de forma consecutiva (al estar diseñado para su uso con cintas). También es el programa del proyecto GNU que se utiliza para gestionar y crear los ficheros del mismo nombre, permitiendo además realizar una compresión sobre el archivado.

Tubería: (o pipeline) Conexión que se puede realizar entre procesos, permitiendo comunicarlo y utilizar la salida de un proceso como entrada del siguiente.

UML: (Unified Modelng Language) Lenguaje de modelado estandarizado para la ingeniería del software orientada a objetos. El estándar fue creado y actualmente es mantenido por el Object Management Group http://www.omg.org/.

Unix: Sistema operativo multitarea y multiusuario desarrollado en 1969 en los laboratorios Bell de AT&T. También se suele denominar como sistemas Unix a todos aquellos sistemas operativos que, aunque no son el Unix de AT&T, son compatibles con el original: ofrecen la misma interfaz (POSIX) y su funcionamiento es similar. En el ámbito de éste proyecto el término "Unix" se refiere a sistemas compatibles con Unix.

Vi: (Visual) Aplicación para la edición de texto. Su principal característica es que ofrece al usuario distintos modos, que permiten realizar distintos tipos de operaciones con el texto. En concreto tiene el modo "Comandos", en el que se pueden hacer operaciones sobre el texto en conjunto, y el modo "Inserción", que permite editar el texto.

Virtualización: Creación de una capa de abstracción entre el hardware de una máquina y un sistema operativo huésped, que crea una versión virtual de un dispositivo. Esta capa software se encarga de repartir los recursos físicos de la máquina, entre las máquinas virtuales ejecutadas sobre la máquina anfitrión real.

Virtualización completa: Tipo de virtualización en el que se simula una máquina completa que permite alojar el sistema operativo huésped sin necesidad de modificarlo.

Wiki: Sitio web que permite la creación y edición, con un navegador web, de cualquier número de páginas interrelacionadas, mediante el uso de un lenguaje de marcado simplificado o un editor de texto. Normalmente su uso y desarrollo es colaborativo por múltiples usuarios.

XMPP: (Extensible Messaging and Presence Protocol) Protocolo abierto para la comunicación mediante mensajería instantánea, creado por la comunidad Jabber en 1999. Actualmente soporta, desde funciones de mensajería de texto, hasta transferencia de fichero o llamadas de voz sobre IP. XMPP es un sistema abierto, de forma que cualquiera puede crear su propio servidor de este servicio de mensajería y conectarlo a la red XMPP.

Capítulo IX

Referencias

ACHOUR, Mehdi; BETZ, Friedhelm; DOVGAL, Antony; LOPES, Nuno; MAGNUSSON, Hannes; RICHTER, Georg; SEGUY, Damien y VRANA, Jakub. *Manual de PHP* [en línea]. Ed. Philip Olson. 2011. [Consultado el 03 Octubre 2011]. Disponible en: <http://www.php.net/manual/es/index.php>.

ARIAS CHAVES, Michael. *Percepción general de la virtualización de los recursos informáticos* [en línea]. Inter Sedes, 2008 - 2009. ISSN: 1409-4746. [Consultado el 04 Octubre 2011]. Disponible en: <http://www.intersedes.ucr.ac.cr/ojs/index.php/intersedes/article/view/214/213>.

BAILEY, D. y KURLAND, N. *A review of telework research: Findings, new directions, and lessons for the study of modern work*. Journal of Organizational Dynamics. 2002, Vol. 28, Págs. 383-400.

BLANCO ROMERO, Asunción. *Teletrabajo, género y territorio. Una comparación entre Cataluña, Ardèche y Québec* [en línea]. Dirección: Gemma Cànoves Valiente. Universitat Autònoma de Barcelona, Departament de Geografía, 2005. [Consultado el 27 Septiembre 2011]. Disponible en: <http://tesisenred.net/bitstream/handle/10803/4960/abr1de1.pdf?sequence=1>.

CIMARRA CARDENAL, Arturo. *Teletrabajo ¿la tendencia del siglo XXI?* [en línea]. Equipos y Talento. 2005 [Consultado el 25 Septiembre 2011]. Disponible en: <http://www.equiposytalento.com/tribunas/page-personnel/teletrabajobrla-tendencia-del-siglo-xxi>.

CULEBRO JUÁREZ, Montserrat; GÓMEZ HERRERA, Wendy Guadalupe y TORRES SÁNCHEZ, Susana. *Software libre versus software propietario. Ventajas y desventajas.* 2006.

EASTEP, Tom. *Shorewall Introduction* [en línea]. 2009. [Consultado el 13 Octubre 2011]. Disponible en: <http://shorewall.net/>.

FREE SOFTWARE FOUNDATION. *Definición de Software Libre* [en línea]. 2011a. [Consultado el 28 Septiembre 2011]. Disponible en: <http://www.gnu.org/philosophy/free-sw.es.html>.

FREE SOFTWARE FOUNDATION. *¿Qué es el copyleft?* [en línea]. 2011b. [Consultado el 29 Noviembre 2011]. Disponible en: <http://www.gnu.org/copyleft/copyleft.es.html>.

FUERTES DÍAZ, Walter y LÓPEZ DE VERGARA MÉNDEZ, Jorge Enrique. *Evaluación de plataformas de virtualización para experimentación de servicios en redes IP* [en línea]. [s.f.]. [Consultado el 04 Octubre 2011]. Disponible en: <http://biblioteca.espe.edu.ec/upload/Revista_WFuertes_JLopez_de_Vergara_Final.pdf>.

GALÁN MÁRQUEZ, Fermín y FERNÁNDEZ CAMBRONERO, David. *VNUML: Una herramienta de virtualización de redes basada en software libre* [en línea]. 2004. [Consultado el 04 Octubre 2011]. Disponible en: <http://jungla.dit.upm.es/~vnuml/papers/OSWC_2004.pdf>.

GARCÍA ALFARO, Joaquin. *Mecanismos de prevención*. 2004.

GARCÍA CALAHORRO, Alberto. *Estudio de rendimiento y funcionalidad sobre diferentes soluciones de virtualización* [en línea]. Dirección: Josep Prieto Blázquez. Universitat Autònoma de Barcelona, Departament d'Enginyeria de la Informació i de les Comunicacions, 2009. [Consultado el 04 Octubre 2011]. Disponible en: <http://www.recercat.net/bitstream/2072/48088/1/PFC_AlbertoGarciaCalahorro.pdf>.

GIL BÁZQUEZ, Sergio. *Plataforma de virtualizacion para teletrabajadores de la UC3M desarrollada con Software Libre*. Dirección: Vicente Palacios Madrid. Universidad Carlos III de Madrid, Departamento de Informática, 2011.

HAZEL, Philip. *The Exim SMTP mail server: official guide for release 4* [en línea]. Cambridge: UIT Cambridge, 2003. ISBN: 0954452909. [Consultado el 05 Octubre 2011]. Disponible en: <http://books.google.com/books?hl=es&lr=&id=foCRVaMeRMgC&pgis=1>.

IBARRA LEMAS, M.C. Francisco. *Firewalls en Linux*. 2006.

JAM SOFTWARE GMBH. *JAM Software - HeavyLoad - Online Hilfe* [en línea]. [Trier], 2011. [Consultado el 12 Octubre 2011]. Disponible en: <http://www.jam-software.com/heavyload/manual.php>.

JONES, Kathryn M. y GONZÁLEZ, Esteban. *Secretos del VM: Virtualización y Drivers* [en línea]. 2008. [Consultado el 04 Octubre 2011]. Disponible en: <http://www.di-mare.com/adolfo/cursos/2008-2/pp-VM.pdf>.

KOCJAN, Wojciech. *Learning Nagios 3.0 A detailed tutorial to setting up, configuring, and managing this easy and effective system monitoring software*. Ed. Manish Sapariya. Birmingham: Packt Publishing Ltd, 2009. ISBN: 9781847195180.

KURLAND, N. y BAILEY, D. *When workers are here, there, and everywhere: A discussion of the advantages and challenges of telework*. Organizational Dynamics. 1999, Vol. 28, Págs. 53-68.

NAGIOS ENTERPRISES. *About Nagios* [en línea]. 2011. [Consultado el 20 Octubre 2011]. Disponible en: <http://www.nagios.org/about>.

NALLEY, David. *Monitorización de la red con Linux, El vigilante*. Linux Magazine. 2009, Vol. 50.

NELSON, Steven. *Pro Data Backup and Recovery*. Apress, 2011. Pág. 294. ISBN: 9781430226628.

OPEN SOURCE INITIATIVE. *La Definición de Código Fuente Abierto* [en línea]. 2003. [Consultado el 29 Noviembre 2011]. Disponible en: <http://www.free-soft.org/mirrors/www.opensource.org/docs/osd-spanish.php>.

OPENIDEAS.INFO. *Pandora 3.0:Documentation es:Introduccion* [en línea]. 2011. [Consultado el 25 Octubre 2011]. Disponible en: <http://www.openideas.info/wiki/index.php?title=Pandora_3.0:Documentation_es:Introduccion>.

PACHEV, Alexander y PACHEV, Sasha. *Understanding MySQL internals* [en línea]. O'Reilly Media, Inc., 2007. Vol. 0, Pág. 234. ISBN: 0596009577. [Consultado el 12 Octubre 2011]. Disponible en: <http://books.google.com/books?id=vz6PcTdo8VUC&pgis=1>.

PROXMOX SERVER SOLUTIONS GMBH. *Proxmox VE* [en línea]. 2011a. [Consultado el 08 Octubre 2011]. Disponible en: <http://www.proxmox.com>.

PROXMOX SERVER SOLUTIONS GMBH. *Proxmox VE Wiki* [en línea]. 2011b. [Consultado el 12 Octubre 2011]. Disponible en: <http://pve.proxmox.com/wiki/Main_Page>.

PROYECTO DEBIAN. *Debian -- Acerca de Debian* [en línea]. 2011. [Consultado el 01 Octubre 2011]. Disponible en: <http://www.debian.org/intro/about>.

RAMEY, Chet. *Bash Introduction* [en línea]. 2011. [Consultado el 06 Octubre 2011]. Disponible en: <http://tiswww.case.edu/php/chet/bash/bash-intro.html>.

REESE, George; YARGER, Randy Jay; KING, Tim y WILLIAMS, Hugh E. *Managing and using MySQL* [en línea]. Eds. Andy Oram y Ellen Siever. 2º ed. Sebastopol: O'Reilly Media, Inc., 2002. ISBN: 0596002114. [Consultado el 09 Octubre 2011]. Disponible en: <http://books.google.com/books?hl=es&lr=&id=kVB1wiF87ooC&pgis=1>.

SEOANE PASCUAL, Joaquín; GONZÁLEZ BARAHONA, Jesus M. y ROBLES, Gregorio. *Introducción al software libre*. 2007.

STALLMAN, Richard M. *Software libre para una sociedad libre* [en línea]. Traficantes de Sueños, 2004. ISBN: 84-933555-1-8. [Consultado el 28 Septiembre 2011]. Disponible en: <http://espora.org/biblioweb/sl-ca/slsl/softlibre-1.2.pdf>.

TANENBAUM, Andrew S. *Sistemas operativos modernos* [en línea]. 2º ed. Pearson Educación, 2003. ISBN: 9702603153. [Consultado el 02 Octubre 2011]. Disponible en: <http://books.google.com/books?hl=es&lr=&id=g88A4rxPH3wC&pgis=1>.

THE APACHE SOFTWARE FOUNDATION. *About the Apache HTTP Server Project - The Apache HTTP Server Project* [en línea]. 2011. [Consultado el 04 Octubre 2011]. Disponible en: <http://httpd.apache.org/ABOUT_APACHE.html>.

THE GNU PROJECT. *The GNU Bash Reference Manual* [en línea]. 4.1 ed. 2009. [Consultado el 06 Octubre 2011]. Disponible en: <http://www.gnu.org/s/bash/manual/bash.html#Introduction>.

THE PHP GROUP. *PHP: Listado de Timezones soportados - Manual* [en línea]. 2011. [Consultado el 23 Agosto 2011]. Disponible en: <http://es.php.net/manual/es/timezones.php>.

THIBAUD, Cyril. *MySQL 5: instalación, implementación, administración, programación* [en línea]. Ediciones ENI, 2006. Pág. 464. ISBN: 2746030691. [Consultado el 12 Octubre 2011]. Disponible en: <http://books.google.com/books?id=wY0bHPmW-NUC&pgis=1>.

ZABBIX SIA. *Zabbix 1.8 manual* [en línea]. 2011. [Consultado el 23 Agosto 2011]. Disponible en: <http://www.zabbix.com/documentation/1.8/complete>.

Capítulo X

Anexos

En este capítulo de anexos se incluyen los scripts desarrollados durante la fase de implantación.

Se comenzará con el script más sencillo, que es el que se utilizará por parte de los agentes de Zabbix para comprobar que el firewall está activo en la máquina que se ejecuta.

Los tres últimos scripts son los localizados en los servidores de Teletrabajo, y encargados de la realización del backup. El primero de ellos es el encargado de copiar los ficheros de configuración de las máquinas virtuales entre los nodos del clúster. El segundo es el encargado de realizar, restaurar y mantener las copias de seguridad de las máquinas virtuales, además de prepararlas para la realización del backup corporativo. El último script es el que se utilizará para programar los dos anteriores en el sistema, limitándose ejecutarlos con los parámetros apropiados.

10.1. Script de verificación de estado del firewall

```
#!/bin/bash

function main() {

resultado1=$(iptables -L INPUT 1)
resultado2=$(iptables -L FORWARD 1)
resultado3=$(iptables -L OUTPUT 1)
```

```
longitud1=$(echo -n $resultado1 | wc -m)
longitud2=$(echo -n $resultado2 | wc -m)
longitud3=$(echo -n $resultado3 | wc -m)

suma=$(($longitud1 + $longitud2 + $longitud3 ))

return $suma

}
main
echo $?
```

10.2. Script de backup cruzado

```
#!/bin/bash

if [ $# -eq 1 ]; then
        directorioOrigen="/etc/qemu-server"
        maquina=$(cat /etc/hostname)

        remoto=""
        directorioDestino=""
        numeroCopias=$1

        if [ $maquina == "prox1" ]; then
                remoto="prox2.uc3m.es"
                directorioDestino="/etc/qemu-server/prox2"
        else
                remoto="prox1.uc3m.es"
                directorioDestino="/etc/qemu-server/prox1"
        fi

        respuesta=$(find / -path $directorioDestino)
        if [ ! $respuesta ]; then
                mkdir $directorioDestino
```

```
            fi


            directorioFinal="$directorioDestino/$(date  +%Y_%m_%d-
%H_%M_%S)"
            mkdir $directorioFinal
            scp $remoto:$directorioOrigen/*.conf $directorioFinal/


            copias=( $(ls -tr $directorioDestino) )
            elementos=${#copias[*]}


            while [ $elementos -gt $numeroCopias ]; do
                    rm -r $directorioDestino/${copias[0]}
                    copias=( $(ls -tr $directorioDestino) )
                    elementos=${#copias[*]}
            done
    else
            echo "Este script se encarga de realizar copias de
seguridad en el directorio /etc/qemu-server/<nombre_maquina> de
los ficheros de configuracion de las maquinas virtuales del otro
servidor del cluster. Donde <nombre_maquina> es el nombre del
servidor del que se realizaran las copias."
            echo "Uso: ./crossoverbackup <NUM>"
            echo "     <NUM>: Numero maximo de copias de seguridad
que se guardaran."
    fi
```

10.3. Script de backup de máquinas virtuales

```
    #!/bin/bash


    directorioMaquinas="/etc/qemu-server"
    directorioBackups="/var/lib/vz/snapshot"
    directorioBackupsCorporativo="/var/lib/vz/backup"
    idInicial=$4
    idFinal=$5


    if [ $# -eq 5 ] && [ $1 == "-c" ]; then
```

```
        echo "Iniciando respaldo..."
        echo
        VMID_maquinas=( $(ls $directorioMaquinas | grep .conf) )
        for maquina in ${VMID_maquinas[*]}; do
            if [ ${maquina%.*} -ge $idInicial ] && [ ${maquina%.*}
-le $idFinal ];then
                echo "Maquina ${maquina%.*}:"
                echo
                backupDir=$directorioBackups/${maquina%.*}

backupDirCorporativo=$directorioBackupsCorporativo/${maquina%.*}
                mkdir -p $backupDir
                echo "Realizando copia de seguridad completa..."
                echo
                vzdump --compress --dumpdir $backupDir --maxfiles
$2 --snapshot ${maquina%.*}
                echo
                echo "Copia de seguridad de maquina completada."
                echo
                diaSemana=$(date  +%u)
                if [ $diaSemana -eq 7 ] || [ $diaSemana -eq 1 ];
then
                    mkdir -p $backupDirCorporativo
                    backups=( $(ls -tr --hide=*.log $backupDir) )
                    ultimoBackup=$((${#backups[*]} - 1))
                    rm -f $backupDirCorporativo/*
                    mv $backupDir/${backups[$ultimoBackup]}
$backupDirCorporativo/${backups[$ultimoBackup]}
                    ln -s
$backupDirCorporativo/${backups[$ultimoBackup]}
$backupDir/${backups[$ultimoBackup]}
                fi
            fi
        done
        echo "Respaldo finalizado."
        echo "Comprobando copias obsoletas..."
        tiempo_actual=$(date +%s)
        tiempo_limite=$(($3 * 86400))
        fecha_limite=$(($tiempo_actual - $tiempo_limite))
        VMID_maquinas=( $(ls $directorioBackups) )
```

```
        for directorio in ${VMID_maquinas[*]}; do
            fecha_directorio=$(date -r
$directorioBackups/$directorio +%s)
            if [ $fecha_directorio -lt $fecha_limite ]; then
                rm -r $directorioBackups/$directorio
                echo "Directorio obsoleto eliminado:
$directorioBackups/$directorio"
            fi
        done
        VMID_maquinas=( $(ls $directorioBackupsCorporativo) )
        for directorio in ${VMID_maquinas[*]}; do
            fecha_directorio=$(date -r
$directorioBackupsCorporativo/$directorio +%s)
            if [ $fecha_directorio -lt $fecha_limite ]; then
                rm -r $directorioBackupsCorporativo/$directorio
                echo "Directorio obsoleto eliminado:
$directorioBackupsCorporativo/$directorio"
            fi
        done
        echo "Tarea finalizada."
    elif [ $# -eq 3 ] && [ $1 == "-r" ]; then
        VMID_backups=( $(ls $directorioBackups) )
        VMID="KO"
        for VMID_backup in ${VMID_backups[*]}; do
            if [ $VMID_backup == $2 ]; then
                VMID="OK"
                backups=( $(ls -tr --hide=*.log
$directorioBackups/$VMID_backup) )
                BACKUP="KO"
                for backup in ${backups[*]}; do
                    if [ $backup == $3 ]; then
                        echo "Iniciando restauracion de copia de
seguridad..."
                        echo
                        BACKUP="OK"

restauracion=$directorioBackups/$VMID_backup/$backup
                        echo "Parando maquina virtual..."
                        echo
                        qm stop $VMID_backup
```

```
                        echo "Eliminando maquina virtual..."
                        echo
                        qm destroy $VMID_backup
                        echo "Restaurando copia de seguridad..."
                        echo
                        qmrestore $restauracion $VMID_backup
                        echo
                        echo "Iniciando maquina virtual..."
                        echo
                        qm start $VMID_backup
                    fi
                done
            fi
        done
        if [ $VMID == "KO" ]; then
            echo -n "No existe ninguna maquina con el
identificador "
            echo -n $2
            echo ". Ejecute solo con \"-r\" para ver el conjunto
de maquinas disponibles."
        else
            if [ $BACKUP == "KO" ]; then
                echo -n "No existe ninguna copia de seguridad para
la maquina "
                echo -n $2
                echo -n " con el nombre "
                echo -n $3
                echo ". Ejecute indicando unicamente la maquina
virtual para ver el conjunto de copias de seguridad disponibles."
            else
                echo "Restauracion de copia de seguridad
terminada."
            fi
        fi
    else
        if [ $# -eq 0 ]; then
            echo "Este script realiza una copia de seguridad de
las maquinas virtuales indicadas con el flag -c o restaura una
copia de seguridad concreta para alguna maquina virtual dada con
el flag -r."
```

```
            echo
        fi
        echo "Uso: ./vmbackup [-c <NUM> <DIAS> <MAQ_INICIO>
<MAQ_FIN> | -r [<VMID> [<BACKUP>]]]"
        if [ $# -eq 0 ]; then
            echo "     -c: Genera una copia de seguridad para cada
maquina virtual."
            echo "         <NUM>: Numero de copias de seguridad
soportadas."
            echo "         <DIAS>: Numero de dias que se
almacenaran las copias obsoletas de una maquina."
            echo "         <MAQ_INICIO>: Numero identificador de
la primera maquina de la que se realizara copia."
            echo "         <MAQ_FIN>: Numero identificador de la
ultima maquina de la que se realizara copia."
            echo "     -r: Restaura la copia de seguridad <BACKUP>
para la maquina <VMID>. ATENCION: La maquina <VMID> sera parada
antes de restaurar la copia de seguridad y se arrancara cuando el
proceso haya terminado."
            echo "         <VMID>: Identificador de la maquina que
se quiere restaurar."
            echo "         <BACKUP>: Nombre de la copia de
seguridad a restaurar."
        fi
        echo


        if [ $# -eq 1 ] && [ $1 == "-r" ]; then
            echo "Maquinas disponibles a las que restaurar una
copia de seguridad:"
            ls -1 $directorioBackups
        fi


        if [ $# -eq 2 ] && [ $1 == "-r" ]; then
            VMID_backups=( $(ls $directorioBackups) )
            VMID="KO"
            for VMID_backup in ${VMID_backups[*]}; do
                if [ $VMID_backup == $2 ]; then
                    VMID="OK"
                    echo -n "Copias de seguridad disponibles para
la maquina "
                    echo -n $2
                    echo ":"
```

```
                    ls -1tr --hide=*.log
$directorioBackups/$VMID_backup
                fi
            done
            if [ $VMID == "KO" ]; then
                echo -n "No existe ninguna maquina con el
identificador "
                echo -n $2
                echo "."
            fi
        fi
    fi
```

10.4. Script de backup completo

```
    #!/bin/bash


    MAQUINA_INICIAL=100
    MAQUINA_FINAL=999
    NUMERO_DIAS=14
    NUMERO_COPIAS_CONF=2
    NUMERO_COPIAS_MAQUINAS=2


    date
    /usr/local/bin/vmtools/crossoverbackup $NUMERO_COPIAS_CONF
#Realiza copias de seguridad de los ficheros de configuracion de
las maquinas del otro servidor
    /usr/local/bin/vmtools/vmbackup -c $NUMERO_COPIAS_MAQUINAS
$NUMERO_DIAS $MAQUINA_INICIAL $MAQUINA_FINAL #Realiza copias de
seguridad de las maquinas de este servidor en directorio local
    date
```